



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ŘÍDICÍ SYSTÉM PRO 3D TISKÁRNU

3D PRINTER CONTROL SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Radim Vítek

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jakub Arm

BRNO 2017

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Radim Vítek

ID: 173776

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Řídicí systém pro 3D tiskárnu

POKYNY PRO VYPRACOVÁNÍ:

- 1) Proveďte rešerši rozběhových profilů krokových motorů.
- 2) Proveďte rešerši interpolačních algoritmů.
- 3) Navrhněte a realizujte rozšiřující DPS pro platformu Nucleo F411 pro připojení sensorů a aktorů.
- 4) Navrhněte a implementujte firmware v jazyce C/C++ pro řízení 3D tiskárny pomocí vybraných G-kódů pro procesor STM32F4 zabudovaný v kitu Nucleo F411.
- 5) Vytvořte PC aplikaci, která bude ovládat 3D tiskárnu po sběrnici RS485, tedy zobrazovat aktuální stav a posílat příkazy.
- 6) Systém otestujte a vyhodnoťte jeho funkčnost.

DOPORUČENÁ LITERATURA:

Brian W. Kernighan, Dennis M. Ritchie. The C Programming Language. Prentice Hall Professional Technical Reference, 1988. ISBN:0131103709.

Termín zadání: 6.2.2017

Termín odevzdání: 29.5.2017

Vedoucí práce: Ing. Jakub Arm

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá způsoby řízení 3D tiskáren. Zejména krokovými motory spolu s jejich rozběhovými profily. Dále se pojednává o interpolačních algoritmech. Pozornost je také věnována návrhu řídicí desky pro 3D tiskárnu, postavenou na kitu Nucleo F411. Poslední část rozebírá implementaci řídicího firmwaru.

KLÍČOVÁ SLOVA

CNC, Nucleo F411, interpolační algoritmy, rozběhové profily, krokové motory, 3D tiskárna, S-křivka

ABSTRACT

This thesis describes ways to controll 3D printers. Especially controlling stepper motors with their motion profiles. Furthermore discusses about interpolation algorithms. Attention is also paid to the design of the control board for the 3D printer built on kit Nucleo F411. The last part describes implementation of controll firmware.

KEYWORDS

CNC, Nucleo F411, interpolation algorithms, motion profiles, stepper motors, 3D printer, S-shaped curve

VÍTEK, Radim. *Řídicí systém pro 3D tiskárnu*. Brno, 2017, 59 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Jakub Arm,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Řídicí systém pro 3D tiskárnu“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce panu Ing. Jakubu Armovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 3D tisk	11
1.1 FDM - Fused Deposition Modeling	11
1.2 SLS - Selective Laser Sintering	12
1.3 SLA - Stereolitografie	12
1.4 LOM - Laminated Object Manufacturing	12
2 Krokové motory	13
2.1 Princip činnosti	13
2.1.1 Krokový motor s permanentními magnety	13
2.1.2 Krokový motor s variabilní reluktancí	14
2.1.3 Hybridní krokový motor	14
2.2 Řízení	15
3 Rozběhové profily	16
3.1 Obdélníkový profil	16
3.2 Lichoběžníkový profil	16
3.3 Rozběhový profil po S-křivce	17
4 Interpolační algoritmy	19
4.1 DDA algoritmus	19
4.2 Bresenhamův algoritmus	21
4.3 Algoritmus schodovité aproximace	21
4.4 Algoritmus přímého vyhledávání	22
4.5 NURBS interpolace	23
4.6 Shrnutí	24
5 Návrh řídicí desky	25
5.1 Sběrnice RS-485	25
5.2 Výstupní obvody	26
5.3 Vstupní obvody	26
5.4 Obvody pro motory	27
5.5 Napájecí obvod	28
5.6 Návrh DPS	28
6 Struktura řídicího firmware	30
6.1 Blok komunikace s PC	30

6.1.1	Komunikační protokol	30
6.1.2	Způsob komunikace	32
6.1.3	Implementace komunikace	32
6.2	Parser a interpret	33
6.3	Interpolace	34
6.4	Krokování	34
6.5	Rozběhový profil	35
6.6	Obsluha periférií	37
7	PSD regulátory	39
7.1	Návrh regulátoru pro extrudér	39
7.2	Návrh regulátoru pro tiskovou podložku	42
7.3	Shrnutí	43
8	PC aplikace	44
8.1	Popis ovládací aplikace	44
8.2	Implementace PC aplikace	45
8.3	Činnost aplikace	45
9	Zhodnocení funkčnosti	47
10	Závěr	48
	Literatura	49
	Seznam symbolů, veličin a zkratk	51
	Seznam příloh	52
A	Obsah přiloženého CD	53
B	Vybrané řídicí příkazy	54
C	Trajektorie kruhových interpolačních algoritmů	55
D	Schéma řídicí desky	56
D.1	Schéma řídicí desky hlavní část	56
D.2	Schéma řídicí desky vstupní obvody	57
D.3	Schéma řídicí desky výstupní obvody	58
D.4	Schéma řídicí desky drivery	59

SEZNAM OBRÁZKŮ

2.1	Zjednodušený model krokového motoru [14]	13
2.2	Řez v ose hybridního krokového motoru [11]	14
3.1	Lichoběžníkový rozběhový profil a profil po S křivce	17
4.1	Princip DDA algoritmu [9]	20
4.2	Vývojový diagram algoritmus přímého vyhledávání (převzato z [4])	23
5.1	Schéma zapojení budiče sběrnice RS-485	25
5.2	Schéma zapojení výstupních obvodů	26
5.3	Schéma zapojení vstupních obvodů	27
5.4	Schéma zapojení motorových driverů	28
5.5	Schéma zapojení napájecího zdroje	28
5.6	Výsledná osazená DPS	29
6.1	Blokové schéma řídicího firmware	30
6.2	Detail rozběhu	35
6.3	Průběh rychlostí S-křivky	36
6.4	Charakteristika termistoru s naznačenou interpolací	38
7.1	Naměřená a aproximovaná normovaná přechodová charakteristika extrudéru	40
7.2	Přechodová charakteristika řízeného extrudéru	41
7.3	Naměřená a aproximovaná normovaná přechodová charakteristika tiskové podložky	42
7.4	Přechodová charakteristika řízené tiskové podložky	43
8.1	Vzhled vytvořené ovládací aplikace	44
9.1	Rozlišení měření teploty pro rezistor o hodnotě $100k\Omega$ (modrá) a $5k\Omega$ (oranžová)	47
C.1	Výsledné trajektorie kruhových algoritmů	55

SEZNAM TABULEK

4.1	Tabulka pravidel schodovité aproximace (převzato z [4])	22
4.2	Vlastnosti interpolačních algoritmů	24
6.1	Struktura rámce	31
6.2	Typy dat	31
6.3	Struktura stavových dat	32
B.1	Vybrané podporované kódy a jejich význam	54

ÚVOD

3D tiskárny jsou počítačem řízené stroje, které umožňují vytvářet levné prototypy výrobků. 3D tiskárny se stávají stále dostupnějšími díky open-source projektům. Jejich konstrukce může být velmi jednoduchá a cenově dostupná.

Práce se zabývá způsoby řízení 3D tiskáren typu FDM. Zejména samotnou implementací řídicího algoritmu a návrhem řídicí desky.

Práce v první části pojednává obecně o problematice 3D tiskáren a jejich typech. Dále se zabývá krokovými motory, jejich řízením a problémy, které jsou s nimi svázány. Zvláštní pozornost je věnována ztrátám kroku a metodám, které ztrátám kroku zabráňují. Metodám zabráňujícím ztrátám kroku tzv. rozběhovým profilům je věnována jedna z kapitol.

Dále se práce zabývá interpolačními algoritmy, které jsou používány pro řízení CNC strojů. Jsou popsány základní i některé pokročilé metody interpolací.

Druhá část řeší návrh rozšiřující desky pro kit STM32F411-Nucleo. Zabývá se jednotlivými obvody, které jsou nutné pro správnou činnost 3D tiskárny. Jedná se o vstupy, výstupy nebo např. obvod zajišťující řízení motorů.

Třetí část práce se zabývá implementací řídicího firmwaru pro 3D tiskárnu. Zabývá se implementací interpolačního algoritmu a rozběhového profilu. Dále je popsán způsob regulace teplot extrudéru a tiskové podložky a také komunikace s nadřazeným systémem. Na konci je popsána aplikace pro PC, která zajišťuje komunikaci s tiskárnou.

1 3D TISK

3D tisk je aditivní technologie vytváření 3D objektů, za pomoci speciálních strojů, tzv. 3D tiskáren. 3D tisk se řadí mezi tzv. rapid-prototyping technologie, které slouží pro rychlou výrobu plně funkčních prototypů. Princip 3D tisku je založen na postupném vrstvení materiálu do požadovaného tvaru. Existuje několik principů tisků.

1.1 FDM - Fused Deposition Modeling

FDM je založena na principu postupného vrstvení taveniny plastu. Tiskovým materiálem je struna z termoplastu, nejčastěji z ABS, který je, v roztaveném stavu, vytlačován extrudérem do podoby tenkého vlákna. Vláknem je nanášeno na vyhřívanou podložku. Každá vrstva tvarově odpovídá horizontálnímu řezu tisknutého objektu. Po vytisknutí vrstvy se tisková hlava zvedne na úroveň další vrstvy a celý proces nanášení plastu se opakuje, dokud není vytištěn celý objekt. [19]

FDM tiskárny se většinou konstruují jako stroje kartézského typu (tříosé pravoúhlé) s XY kinematikou, kde v osách X a Y se pohybuje extrudér a v ose Z tisková podložka a kde je každá osa poháněna právě jedním motorem. Kartézské tiskárny mají výbornou tuhost a přesnost. Jejich řízení je nenáročné na výpočetní výkon, často bývá implementováno na 8-bit mikrokontrolérech.

Kromě kartézského typu s XY kinematikou je možné se setkat také s typem využívající delta kinematiku nebo H-bot kinematiku. Delta tiskárny mívají vyšší rychlost posuvu, avšak za cenu nižší přesnosti. Z důvodu velké konstrukční výšky nemají takovou tuhost jako kartézské typy. Také řízení vyžaduje vyšší výpočetní výkon. [8]

Kinematika H-bot využívá součinnosti dvou motorů pro pohyb v jedné ose. Výhodou je, že oba motory jsou umístěny v rozích statické části konstrukce. Naopak se zde vyskytuje problém s pružností dlouhého řemenu, který je pro použití H-bot kinematiky nutností. Další nevýhodou může být nevyrovnaný průběh pohybu pouze v jedné ose, kdy musí oba motory vykonat krok ve stejný okamžik. Nepřesnosti v časování se mohou projevit pohyby po pilovité trajektorii.

Technologie FDM je masově rozšířená zejména díky open-source projektu RepRap, dobré dostupnosti spotřebního materiálu a jeho nízké ceně. FDM tiskárny umožňují tisk z více materiálů. Výška vrstvy nanášeného materiálu se obvykle pohybuje okolo 0,25 mm. [13]

1.2 SLS - Selective Laser Sintering

Technologie založená na laserovém spékání práškového materiálu. Práškový materiál je nanášen v tenkých vrstvách a laserem zapékán v místech, kde má vzniknout pevné těleso. Po zapečení jedné vrstvy se celý výtisk i s nezapečeným práškem posune směrem dolů a překryje se novou vrstvou materiálu. Proces se opakuje, dokud není dokončen celý výtisk. Materiál bývá nejčastěji plast, ale je možné také tisknout například z kovu, nebo jiných materiálů.

SLS tiskárny se konstruují jako kartézské s pohyblivým laserem. Technologie je využívána pro výtisky, které není možné vyrobit jinak. Jedná se o velice drahou technologii jak pořízovací cenou samotné tiskárny tak i provozními náklady.

Výhodou metody je, že je možné vytisknout libovolný tvar s libovolnou orientací. Je to způsobeno tím, že výtisk je vždy ze všech stran obklopen práškem, který jej stabilizuje a podpírá. Z toho plyne, že při tisku není potřeba vytvářet žádné podpory. Přesnost SLS tiskáren se pohybuje okolo 0,1 mm. [20]

1.3 SLA - Stereolitografie

Technologie, při níž je pryskyřice po vrstvách vytvrzována pomocí světla. Výhodou SLA je vysoká přesnost, která je teoreticky dána velikostí molekuly. Prakticky se však dosahuje přesnosti asi 25 μm . Další výhodou je možnost tisku z průhledných materiálů, což u ostatních typů tiskáren nejde. U FDM je sice možné použít průhledný materiál, ale výtisk plně průhledný nebude, protože ve výtisku jsou vzduchové mezery mezi jednotlivými vlákny, které lámou světlo. Tento problém u SLA odpadá. [19, 16]

Naopak nevýhodou je relativně vysoká cena samotné tiskárny i pryskyřic. Vzhledem k tomu, že není potřeba tak výkonného zdroje světla jako u SLS, SLA tiskárny se stávají cenově dostupnějšími.

1.4 LOM - Laminated Object Manufacturing

Funguje na principu lepení plátů tiskového materiálu na sebe. Každá vrstva je ořezána na požadovaný tvar a přilepena k předchozí. Tímto způsobem vzniká výsledný výtisk. Tisknutím vzoru na jednotlivé vrstvy je možné získat model v libovolné barvě. Možnost barevného tisku je asi největší výhodou, protože přesnost metody je srovnatelná nebo spíše nižší než FDM.

Tisknout je možné z téměř libovolného materiálu, který je možné laminovat ve vrstvách. Typicky se však tiskne z papíru. LOM není příliš efektivní metodou, protože z každé vrstvy je skutečně využita jen malá část. [19]

2 KROKOVÉ MOTORY

Krokový motor je synchronní elektrický stroj s nespojitým pohybem. Hřídel se pohybuje vždy z jedné stabilní polohy do druhé v tzv. krocích. Stator je složený z několika párů cívek, navinutých na ocelovém jádře opatřeném výstupky. V závislosti na typu motoru se rotor skládá z železného jádra, permanentního magnetu, nebo permanentního magnetu v železném jádře [14].

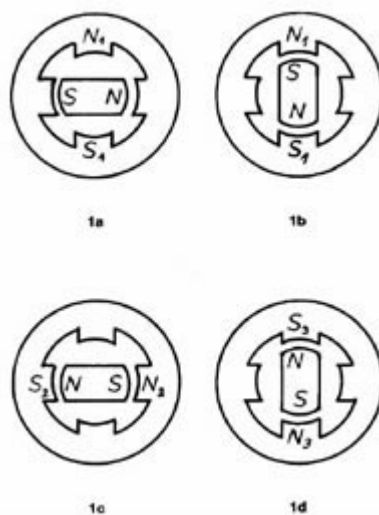
2.1 Princip činnosti

Existují tři základní typy krokových motorů:

- motor s permanentním magnetem,
- motor s variabilní reluktancí,
- hybridní motor.

2.1.1 Krokový motor s permanentními magnety

Motor s permanentními magnety je nejjednodušším typem krokového motoru. Protéká-li jednou z cívek proud, vytvoří se magnetické pole, které natočí rotor. Postupným přepínáním cívek je vyvolán otáčivý pohyb rotoru. Na obrázku 2.1 je znázorněn zjednodušený model krokového motoru s permanentními magnety.



Obr. 2.1: Zjednodušený model krokového motoru [14]

Tento motor se používá pro svůj velký moment a nízkou cenu. Nevýhodou je nižší rychlost a nízké rozlišení, které typicky bývá mezi 30° a 90°. [12]

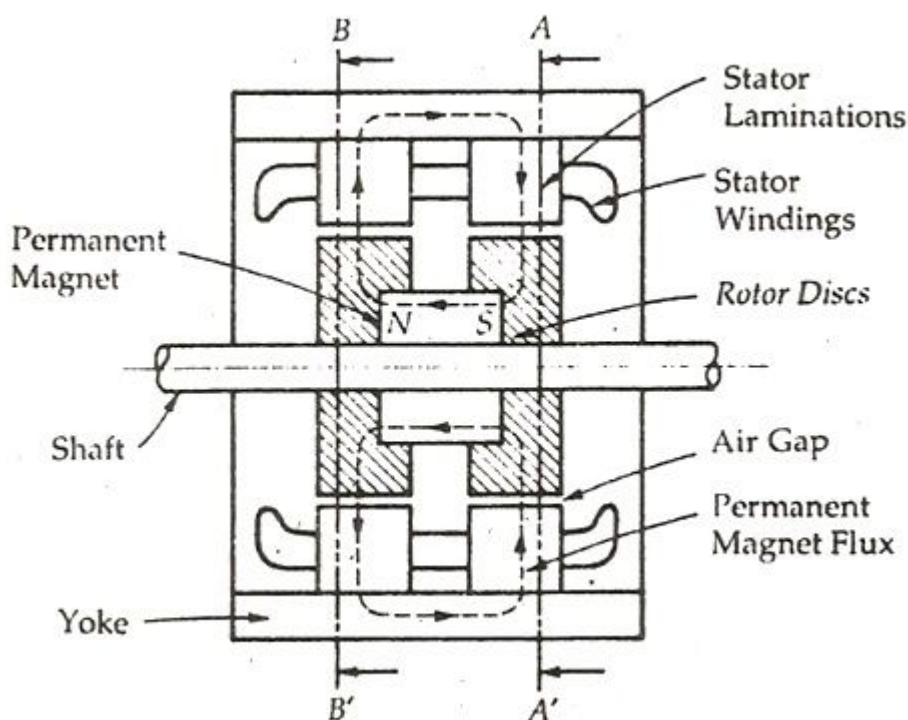
2.1.2 Krokový motor s variabilní reluktancí

Rotor tohoto typu krokového motoru neobsahuje magnet, ale pouze výstupky z magneticky měkkého železa. Při vybuzení cívky je přitahován nejbližší výstupek k pólovému nástavci. Protože rotor nevytváří vlastní magnetické pole, je síla, kterou na sebe působí rotor a stator menší, než v případě krokového motoru s permanentními magnety.

Motor má malý moment, ale může dosahovat vysokých otáček. Malý moment je důvod, proč se tento typ používá jen velice zřídka.

2.1.3 Hybridní krokový motor

Spojuje v sobě vlastnosti ostatních typů krokových motorů. Rotor se skládá ze dvou disků na jejichž obvodu jsou výstupky. Každý z disků má opačnou polaritu magnetického pole danou axiálním magnetem viz obrázek 2.2. Stator je vybaven nejčastěji osmi pólovými nástavci s navinutou cívkou. Vybudí-li cívka severní pól na nástavci, přitáhne se nejbližší výstupek s jižním pólem. Postupným přepínáním cívek s různou polaritou se vyvolá rotační pohyb. [15]



Obr. 2.2: Řez v ose hybridního krokového motoru [11]

Tento typ motoru je nejpoužívanější, kvůli přednostem týkajících se momentu, rychlosti a také rozlišení. Běžně vyráběné hybridní motory mají 200–500 kroků na

otáčku, což odpovídá úhlu $0,72^\circ$ – $1,8^\circ$. Nevýhodou je, že jsou oproti motorům s permanentními magnety dražší.[11]

2.2 Řízení

Krokové motory je možné řídit unipolárně nebo bipolárně. Při unipolárním jednofázovém řízení prochází proud vždy jednou cívkou. Při unipolárním dvoufázovém řízení prochází proud dvěma sousedními cívkami. Při bipolárním jednofázovém řízení proud navíc prochází protější cívkou, nebo dvěma cívkami při řízení dvoufázovém, tak aby magnetická síla působící na rotor byla dvojnásobná.

Základní rozlišení motoru lze zvýšit na dvojnásobek pomocí řízení s polovičním krokem, kde je více aktivních cívek v jeden okamžik. Tím je zajištěno, že rotor zůstane natočen mezi dvěma základními pozicemi. Nebude-li poměr budicího proudu stejný ve všech cívkách, bude se rotor natáčet do úhlů daných poměrem proudů. Tento způsob řízení se nazývá mikrokrokování. Teoreticky je možné dosáhnout libovolného množství mikrokroků, avšak v praxi se používá maximálně 32 mikrokroků. To znamená, že motor se základním úhlem kroku $1,8^\circ$ může být natáčen s přesností přibližně $3,4'$.

Při velkém rozlišení nastává problém s homogenitou magnetického pole. Nebude-li magnetické pole zcela homogenní, budou se úhly natočení jednotlivých mikrokroků vzájemně lišit. Obecně platí, že čím větší přesnost od motoru požadujeme, tím musí být motor kvalitnější a musí mít větší základní rozlišení. Pro získání velkého rozlišení je vhodnější použít převodovku namísto mikrokrokování.

K regulaci krokových motorů se nejčastěji používá otevřená regulační smyčka, kdy se nezjišťuje skutečná pozice, ale pouze se předpokládá, že motor krok skutečně vykonal. Příklad, kdy motor nemá dost energie, aby krok udělal, se označuje jako ztráta kroku, která při řízení v otevřené smyčce způsobuje stálou chybu.

Tento problém řeší řízení v uzavřené smyčce, kde se zjišťuje aktuální pozice a případná odchylka se koriguje. Použití odměřování přináší mnoho dalších problémů. Jmenovitě se jedná o složitost řídicího algoritmu, cena odměřování nebo náchylnost odměřovacího zařízení na mechanické poškození.

Nejčastějším řešením je použití vhodného rozběhového profilu a předimenzování motoru. V takovém případě je možné využívat řízení v otevřené smyčce bez obav ze ztráty kroku.

3 ROZBĚHOVÉ PROFILY

Rozběhové profily definují průběh rychlosti otáčení během rozběhu resp. doběhu motoru. Použitím rozběhových profilů je možné snížit riziko ztráty kroku na minimum. Další výhodou použití rozběhových profilů je plynulost pohybu. Při použití rozběhových profilů, tzn. plynulejším pohybu, může být konstrukce stroje méně masivní.

Jednou z příčin ztráty kroku je přílišná strmost rozběhu. Veličina, která přímo souvisí se strmostí rozběhu se nazývá ryv. Ryv, v anglické literatuře označován jako jerk, odpovídá druhé derivaci rychlosti. Omezením ryvu na rozumnou úroveň je možné zcela zabránit nežádoucím ztrátám kroku, které vedou ke špatnému pozicování. Vztah ryvu k ostatním veličinám charakterizujícím pohyb je:

$$j = \frac{da}{dt} = \frac{d^2v}{dt} = \frac{d^3s}{dt} \quad (3.1)$$

Podle tvaru rozběhových profilů rozlišujeme tři základní typy. Obdélníkový, lichoběžníkový profil a profil po S-křivce. Každý z uvedených profilů zajišťuje jiné průběhy ryvu a má tedy různý vliv na ztrátu kroku.

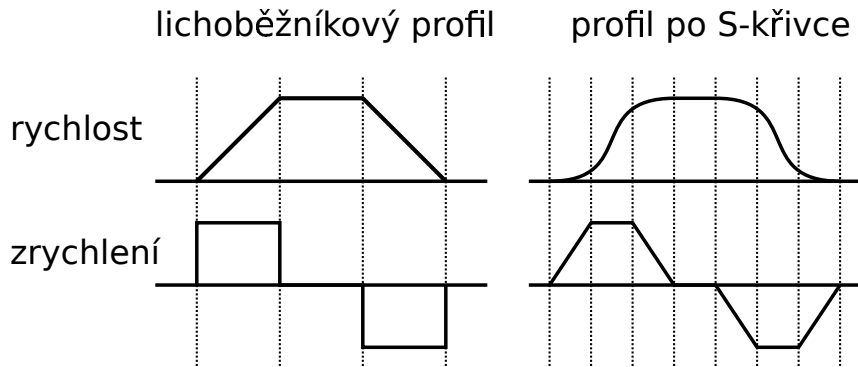
3.1 Obdélníkový profil

Jedná se o přímé zapnutí/vypnutí motoru bez postupného navyšování rychlosti. Tento profil se používá pouze tam, kde motor není při rozběhu zatížen, nebo kde běží trvale beze změn rychlosti. Toto splňují například pohony včetně obráběcích strojů. Vřetena se rozbíhají mimo záběr a po celou dobu řezu udržují konstantní rychlost. Obdélníkový profil se obvykle nepoužívá v kombinaci s krokovým motorem, protože nezabraňuje ztrátě kroku.

V teoretické rovině mají zrychlení i ryv v místě skokové změny rychlosti charakter diracova impulsu. Prakticky je strmost hran omezena a tudíž i diracovy impulzy ztrácejí na své strmosti a profil se spíše podobá lichoběžníkovému profilu s velkou strmostí náběhu resp. doběhu.

3.2 Lichoběžníkový profil

Lichoběžníkový profil, někdy nazývaný jako trapézový profil, je nejjednodušším typem rozběhového profilu, který omezuje ztrátu kroku. V principu se jedná o omezení zrychlení na úroveň, kterou zvládnou motor, měnič i zdroj zpracovat. Výsledkem je dosažení požadované rychlosti s konstantní hodnotou zrychlení, viz levá část obrázku 3.1.



Obr. 3.1: Lichoběžníkový rozběhový profil a profil po S křivce

Lichoběžníkový profil je často používáný pro jednoduchost implementace a nenáročnost na výpočty. Profil omezuje zrychlení na hodnotu, kterou je motor schopen zpracovat, tzn. omezuje ztráty kroku, ale průběh ryvu stále zůstává diracovým impulzem a může způsobovat rázy při změnách rychlosti, nebo, v případě špatného nastavení, i výše zmiňované ztráty kroku. Rázy je možné pozorovat zejména u třískových obráběcích strojů, kde vytváří nevzhledný povrch s velkou drsností. Toto řeší rozběhový profil po S-křivce.[18]

3.3 Rozběhový profil po S-křivce

Profil po S-křivce, při správném použití, zcela odstraňuje ztráty kroku v důsledku setrvačných sil. S-profil také omezuje ryv na předem určenou úroveň. Omezením ryvu lze dosáhnout plynulého rozběhu motoru bez rázů, viz pravá část obrázku 3.1. Změna rychlosti má parabolický průběh, který plynule navazuje na lineární část. Profil po S-křivce lze popsat pomocí rovnic[18]:

$$v(t) = \int_0^t a(t) \cdot dt = a_{avg} \cdot t_a \quad (3.2)$$

$$a_{avg} = \frac{1}{t_a} \int_0^{t_a} a(t) \cdot dt \quad (3.3)$$

$$a_{avg} \leq a_{peak} \leq 2 \cdot a_{avg} \quad (3.4)$$

Při rozběhovém profilu po S-křivce se zrychlení nemění skokově. Tento profil je náročný na výpočty a proto se používá zejména na strojích s výkonnější řídicí jednotkou, nebo tam kde je potřeba zajistit velmi plynulý pohyb.

Výpočetní náročnost vychází z potřeby počítat čas, kdy bude dosaženo požadované rychlosti, a kdy je třeba začít snižovat zrychlení. Výpočty navíc komplikuje

fakt, že doba mezi jednotlivými kroky se mění podle aktuální rychlosti. Proto není možné se rozhodovat pouze podle počtu kroků, ale je nutné skutečné časy počítat.

Změny periody krokování je nutné počítat s určitým předstihem, aby bylo možné správně reagovat na změny rychlosti a naplánovat okamžiky zlomů zrychlení. [7]

V některých strojích se používají pro řízení rychlosti pohybu diskrétní regulátory. Při použití regulátoru, založeném na základním typu PSD, je možné realizovat trapézový rozběhový profil. Profil po S-křivce vyžaduje komplikovanější regulátory.

Ani použití regulátorů neodstraňuje potřebu znát okamžiky zlomů a proto tato metoda není příliš využívána.

4 INTERPOLAČNÍ ALGORITMY

Interpolační algoritmy řídí pohyb CNC stroje v jednotlivých osách podle jim zadaných parametrů tak, aby výsledný pohyb co nejpřesněji kopíroval požadovanou trajektorii. Algoritmů existuje celá řada. Mezi nejvýznamnější patří DDA algoritmus, bresenhamův algoritmus nebo NURBS algoritmy.

Algoritmy je možné klasifikovat podle několika parametrů, zejména podle výpočetní náročnosti dále podle přesnosti nebo typu křivek, které lze pomocí nich interpolovat. Na výpočetní náročnost má nejzásadnější vliv nutnost výpočtů v plovoucí desetinné čárce, kterým se proto snažíme vyhnout. Dále lze klasifikovat podle množství výpočtů. Přesnost je u většiny algoritmů 1 krok, ale některé mohou dosahovat přesnosti až $1/2$ kroku. Vyšší přesnosti než $1/2$ nelze dosáhnout kvůli tomu, že je možné vykonávat pouze celé kroky. Proto se nelze k bodu ležícímu uprostřed mezi kroky přiblížit více.

Křivky, které mohou být interpolovány je celá řada. Základními typy jsou přímka (úsečka) a kružnice, pomocí nichž je možné interpolovat libovolnou složitější křivku. Lineární interpolaci lze využívat i samostatně v případě, že program je pro tuto možnost připraven. Zde se využívá rozkladu křivek na malé elementární úsečky. O rozklad se stará CAM program. Pro obecné NURBS křivky existují tzv. NURBS interpolační algoritmy.

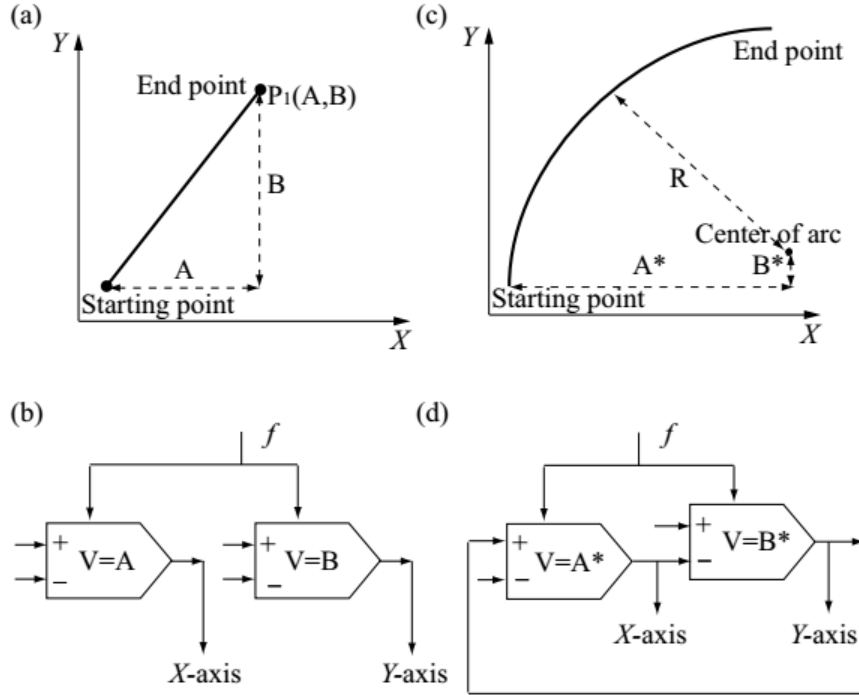
4.1 DDA algoritmus

DDA algoritmus využívá pro svoji činnost numerické integrace. Algoritmus se vykonává cyklicky a výstupem pro každou osu je přetečení integrátoru. Princip činnosti je znázorněn na obrázku 4.1.

Dva integrátory postupně inkrementují svou hodnotu. Maximální hodnota, kterou integrátor naintegruje je V . Po přetečení integrátoru dojde k vykonání kroku na příslušné ose. Algoritmus se vykonává tak dlouho, dokud některá z os nedosáhne požadované pozice. Rychlost pohybu je možné řídit pomocí frekvence, kterou integrátory integrují. Lineární interpolaci lze popsat rovnicemi [9]:

$$\Delta S_x = \left(\frac{f}{2^n} \right) \cdot A \cdot \Delta t \quad (4.1)$$

$$\Delta S_y = \left(\frac{f}{2^n} \right) \cdot B \cdot \Delta t \quad (4.2)$$



Obr. 4.1: Princip DDA algoritmu [9]

Kruhovou interpolaci lze popsat rovnicemi:

$$dX = \omega R \cdot \sin \omega t \cdot dt \quad (4.3a)$$

$$\omega R \cdot \sin(\omega t) \cdot dt = -d(R \cdot \cos \omega t) \quad (4.3b)$$

$$dY = \omega R \cdot \cos \omega t \cdot dt \quad (4.4a)$$

$$\omega R \cdot \cos(\omega t) \cdot dt = d(R \cdot \sin \omega t) \quad (4.4b)$$

Po ukončení cyklu může zbývat necelá část kroku, která způsobuje trvalou odchylku od požadované pozice. V extrémním případě se mohou tyto odchylky po dobu běhu postupně sčítat. Toto je možné vyřešit například tím, že se zbytek kroku rozprostře mezi následující pohyby. Dalším řešením je možnost počítat celkovou odchylku a postupně ji korigovat postupným vykonáváním potřebných kroků.

DDA algoritmus je možné implementovat jak softwarově, tak hardwarově pomocí logických obvodů. Výhodou algoritmu je vysoká přesnost, protože odchylky z principu nevznikají větší než polovina kroku. Další výhodou je vysoká rychlost, snadná implementovatelnost a snadná rozšiřitelnost pro kruhovou interpolaci, kterou lze vidět na obrázku 4.1(d).

DDA interpolace se využívá zejména ve starších strojích v hardwarové podobě a v nových strojích v podobě softwarové. V nových strojích je DDA algoritmus využíván pro interpolaci přímk. Pro kruhovou interpolaci existují vhodnější algoritmy, viz níže. Maximální chyba, které se algoritmus dopustí, je jeden krok.

4.2 Bresenhamův algoritmus

Algoritmus využívá rovinu rozdělenou podle hodnoty směrnice přímky. V dané oblasti musí být směrnice vůči jedné z os menší než 1. Tuto podmínku plní rovina rozdělená na osm částí, tzv. oktanty. Toto rozdělení zajišťuje, že pokud na jedné ose inkrementujeme pozici, na druhé ose se nestane, že by bylo nutné vykonávat více, než jeden jediný krok.

Celý algoritmus tedy funguje tak, že se v ose, na které je posuv rychlejší, vykonávají kroky pravidelně každou periodu řídicí frekvence. Na ose, kde je pohyb pomalejší se počítá odchylka od skutečné pozice a v okamžiku, kdy je odchylka větší než $1/2$ kroku se vykoná celý další krok a odečte se od registru, kde se uchovává velikost odchylky. Implementace se obvykle provádí tak, aby nebyly nutné výpočty v plovoucí desetinné čárce. V úpravě bez výpočtů v plovoucí desetinné čárce se rozhoduje o vykonání kroku na pomalejší ose podle znaménka odchylky.

Využití algoritmu v oblasti řízení CNC strojů není příliš velké, zejména kvůli jeho náročnosti na výpočty. Nejčastěji se bresenhamův algoritmus používá při vykreslování digitální grafiky. Maximální odchylka, které se algoritmus může dopustit, je 1 krok. Bresenhamův algoritmus umožňuje interpolaci přímky i kružnice. [3]

4.3 Algoritmus schodovité aproximace

Algoritmus je určen pro kruhovou interpolaci. Během každého cyklu iterace je počítána aktuální odchylka od požadované pozice na kružnici. Nachází-li se aktuální pozice vně kružnice, vykoná se krok směrem dovnitř a naopak. Dalším parametrem pro určení následujícího kroku je směr interpolace. K určení jestli je aktuální pozice uvnitř nebo vně kruhu slouží rozhodovací rovnice [4].

$$D_k = X_k^2 + Y_k^2 - R^2 \quad (4.5)$$

Jedná se o mírně upravenou rovnici kružnice, kde X_k a Y_k jsou aktuální pozice na jednotlivých osách a R je poloměr kružnice. Výsledkem je aktuální odchylka od požadované pozice na kružnici. Podle D_k se rozhoduje o příštím kroku. Kterým směrem a na které ose se budeme pohybovat, definuje tabulka pravidel.

Tabulka pravidel je nadefinovaná pro oba směry pohybu a pro každý kvadrant tak, aby výsledný pohyb byl kruhový s co nejmenší odchylkou. Tabulka je odvozena z možných výsledků rozhodovací funkce. Například vyjde-li $D_k > 0$ (pozice vně kruhu), pohybujeme se v matematicky kladném směru a nacházíme se v prvním kvadrantu, musí se vykonat pohyb v záporném směru v ose X. Příklad, že je odchylka rovna 0 spadá pod libovolnou z kategorií $D_k > 0$ nebo $D_k < 0$.

Kvadrant	Směr	D<0	D>0
1	záporný	+X	-Y
1	kladný	+Y	-X
2	záporný	+Y	+X
2	kladný	-X	-Y
3	záporný	-X	+Y
3	kladný	-Y	+X
4	záporný	-Y	-X
4	kladný	+X	+Y

Tab. 4.1: Tabulka pravidel schodovité aproximace (převzato z [4])

Algoritmus schodovité aproximace je snadno implementovatelný, ovšem za cenu vyšší výpočetní náročnosti, z důvodu nutnosti počítat druhé mocniny. Jeho maximální odchylka je jeden krok.

4.4 Algoritmus přímého vyhledávání

Upraví-li se rozhodování algoritmu schodovité aproximace tak, aby podporoval pohyb v obou osách současně, sníží se maximální odchylka na $1/2$ kroku. Takto upravený algoritmus je nazýván algoritmem přímého vyhledávání (DSM).

Pro úpravu je nutné rozšířit rozhodovací rovnice. Oproti algoritmu schodovité aproximace jsou tu 3. [4]

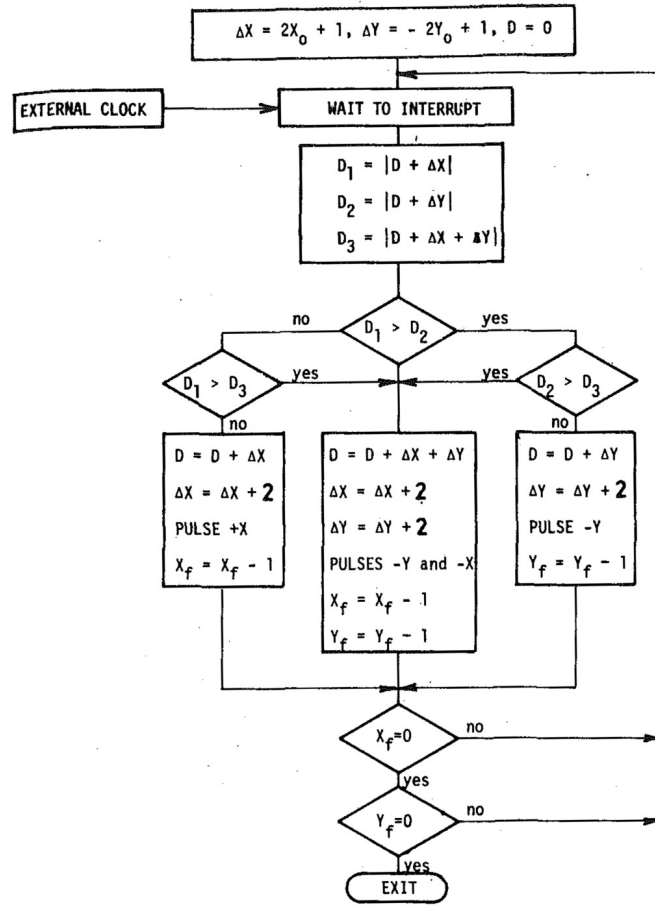
$$D_1 = |D + \Delta X| \quad (4.6)$$

$$D_2 = |D + \Delta Y| \quad (4.7)$$

$$D_3 = |D + \Delta X + \Delta Y| \quad (4.8)$$

Výpočtem parametrů a jejich porovnáním je možné zjistit, kterým směrem je vhodné vykonat krok tak, aby výsledná odchylka od požadované trajektorie byla co nejmenší. Podle této vlastnosti byl algoritmus i pojmenován.

DSM patří mezi nejpřesnější a nejrychlejší algoritmy určené pro kruhovou interpolaci. Celý algoritmus je popsán pomocí vývojového diagramu na obrázku 4.2.



Obr. 4.2: Vývojový diagram algoritmus přímého vyhledávání (převzato z [4])

4.5 NURBS interpolace

NURBS (neuniformní racionální B-spline) křivky jsou aproximační křivky, které nemusí procházet zadanými body [6]. Jsou složeny z racionálních B-spline křivek a zadány pomocí řídicích bodů, jejich vah a stupněm, tímto způsobem lze zadat libovolně tvarovanou křivku. Křivky jsou vytvořeny jako hladké proložení zadaných bodů. Váhy určují jak výrazně se mají jednotlivé body projevit.

Matematický popis NURBS křivek [9]:

$$P(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \quad a \leq u \leq b \quad (4.9)$$

Typ	Algoritmus	Přesnost	Výpočetní náročnost
lineární	DDA	1	velmi malá
	Bresenhamův	1	malá (vyžaduje výpočty s desetinnou čárkou)
kruhový	DDA	1	velmi malá
	Algoritmus schodovité aproximace	1	střední
	Algoritmus přímého vyhledávání	1/2	střední
	NURBS	-	velká

Tab. 4.2: Vlastnosti interpolačních algoritmů

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u \leq u_{i+1} \\ 0 & \text{jinak} \end{cases} \quad (4.10)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Interpolace pomocí NURBS křivek se používá zejména v grafických programech. V interpolátorech CNC strojů se vyskytuje jen zřídka, kvůli své výpočetní náročnosti. Častěji jsou NURBS křivky převáděny na posloupnost krátkých úseček a částí oblouků tak, aby CNC stroj počítal pouze kruhovou a lineární interpolaci.

4.6 Shrnutí

Interpolační algoritmy pro přímky, dosahují téměř stejných výsledků. Liší se zejména v množství a náročnosti výpočtů. Shrnutí vlastností jednotlivých popisovaných algoritmů je v tabulce 4.2 [4].

Kruhové interpolační algoritmy dosahují velice odlišných výsledků. Je to dáno tím, že kruh je podstatně složitější na aproximaci než přímka. Výsledné trajektorie vybraných interpolačních algoritmů lze vidět na obr. C.1. Z obrázku lze vidět, že algoritmus přímého vyhledávání dosahuje nejvyšší přesnosti, naproti tomu DDA algoritmus má nejhladší průběh v oblasti 45°. Při větších poloměrech se nepřesnost algoritmu DDA projevuje výrazněji.

5 NÁVRH ŘÍDICÍ DESKY

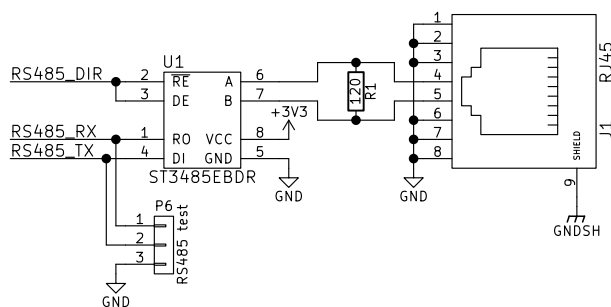
3D tiskárna je řízena platformou Nucleo F411 s mikrokontrolérem STM32F411. Pro modul s mikrokontrolérem bylo nutné navrhnout rozšiřující DPS, která obsahuje všechny periferie nutné k řízení.

Stroj má 3 osy, na každé z os je umístěno referenční čidlo pro určení výchozí pozice. Motory jsou buzeny drivery DRV8825. Dále DPS umožňuje měřit teplotu pomocí tří termistorů, spínat šest tranzistorových výstupů z toho tři výkonové výstupy dimenzované na proud 3 A. Dva výstupy jsou reléové. Komunikace probíhá po sběrnici RS-485.

Na řídicí desce je umístěna 64 kB EEPROM paměť, kde budou uloženy konstanty potřebné pro správnou funkci. Paměť je připojena pomocí sběrnice I²C. Paměť FLASH s kapacitou 8 MB, která je připojena pomocí sběrnice SPI. FLASH paměť je určena pro uložení programu pro obrábění. Kompletní schéma řídicí desky je v příloze D.

5.1 Sběrnice RS-485

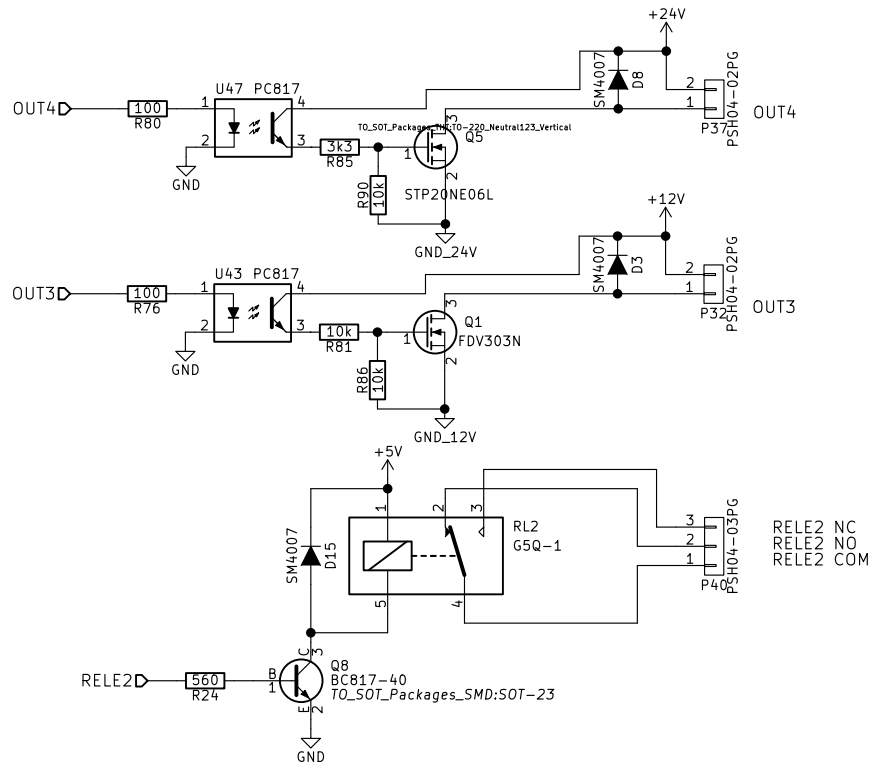
Sběrnice RS-485 je sériová poloduplexní sběrnice vyžívající diferenciální pár pro přenos dat. Pro spojení s PC se využívá stíněný UTP kabel, zakončený konektorem RJ45. Stínění musí být spojeno se stíněním přívodního kabelu napájení. Jako budič byl vybrán obvod ST32485EBDR, který je kompatibilní s 3,3 V logikou, kterou vyžaduje mikrokontrolér. Dále má integrovanou ochranu proti ESD výbojům do napětí 1,5 kV. Proto není použita žádná další ESD ochrana. Rezistor R1 slouží jako zakončovací odpor. Pro usnadnění měření jsou signály RX a TX vyvedeny na kolíkovou lištu.



Obr. 5.1: Schéma zapojení budiče sběrnice RS-485

5.2 Výstupní obvody

Výstupy jsou galvanicky oddělené, aby bylo možné připojit jakýkoli zdroj. Oddělení zajišťuje optočlen PC817. Jeho budící proud je 20 mA při napětí 1,2 V. Hodnota odporu na vstupu vyplývá z ohmova zákona. Tranzistory jsou unipolární MOS-FET s indukovaným kanálem. Tranzistor FDV303N je řízen napětím do 8 V. Maximální proud je 680 mA. Tranzistor STP20NE06L, jehož maximální proud je 20 A, má řídicí napětí maximálně 20 V. Rezistory u tranzistorů tvoří napěťový dělič, omezující maximální řídicí napětí. Jejich hodnoty jsou vybrány tak, aby u tranzistorů nedošlo k průrazu, ale byly dostatečně otevřené. Rezistor připojený na GND navíc zajišťuje vybíjení hradla v klidovém stavu. Dioda připojená paralelně k zátěži slouží jako ochrana proti zpětným proudům. Tranzistory obsahují vlastní substrátovou diodu, kterou je vhodné ještě posílit. Relé je spínáno bipolárním tranzistorem BC817 s maximálním proudem 500 mA.



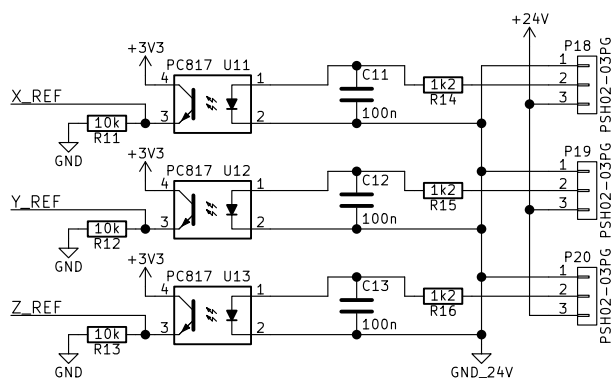
Obr. 5.2: Schéma zapojení výstupních obvodů

5.3 Vstupní obvody

Vstupní obvody jsou galvanicky oddělené pomocí optočlenů PC817. Časová konstanta reakce na vstupní signál je nastavena pomocí RC filtru na $\tau = 120\mu s$. Jako

referenční čidla jsou použity indukčnostní limitní snímače. Aby je bylo možné snadno připojit, jsou konektory třípinové (dva piny napájení a jeden výstupní signál). Na ostatní vstupy budou připojeny pouze kontakty, proto jsou konektory dvoupinové. Bezpečnostní tlačítko musí mít rozpínací kontakty, s ohledem na tento fakt byl navržen obvod tak, aby tlačítko ve všech krizových situacích vyvolalo přerušení činnosti stroje.

Z důvodu nedostatku vývodů na MCU bylo nutné všechny vstupy vyjma bezpečnostního tlačítka zapojit na posuvné registry. Bez použití posuvných registrů by bylo nutné použít 13 vývodů na MCU. Při použití posuvných registrů jsou nutné pouze tři vývody MCU. Posuvné registry 75HC165D byly vybrány po svou kompatibilitu s 3,3 V logikou. Aby bylo možné připojit všech 13 vývodů bylo nutné zapojit dva posuvné registry za sebe.



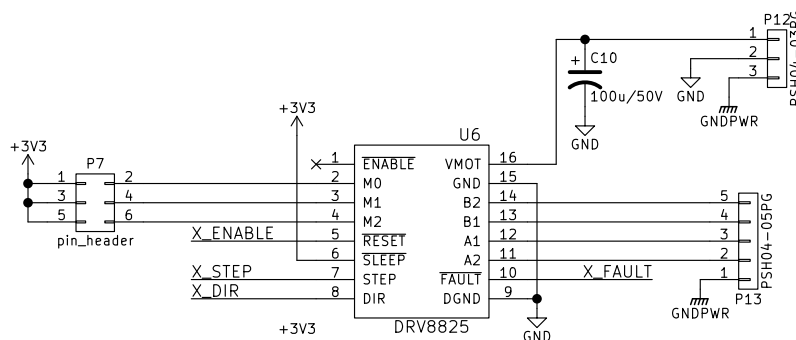
Obr. 5.3: Schéma zapojení vstupních obvodů

5.4 Obvody pro motory

Driver pro krokové motory DRV8825 je kompletní řešení pro řízení krokových motorů. Obsahuje výstupní budiče i řídicí logiku zajišťující například mikrokrokování s rozlišením až 32 mikrokroků, nebo indikaci přehřátí a nadproud. Aby bylo možné nastavovat rozlišení mikrokrokování podle potřeby, jsou vyvedeny potřebné vývody na kolíkovou lištu. Jejich propojováním pomocí jumperů lze nastavit požadované rozlišení. Vývod FAULT potřebuje externí pull-up rezistor, za tímto účelem je využit pull-up rezistor integrovaný do MCU.

Drivery jsou zapojeny podle doporučení výrobce. Odlišností od katalogového zapojení je spojení motorové a digitální země. To není nezbytně nutné, protože modul sám má spojení realizováno. Ale kvůli zachování integrity rozlité mědi, se obě země spojily i na straně desky.

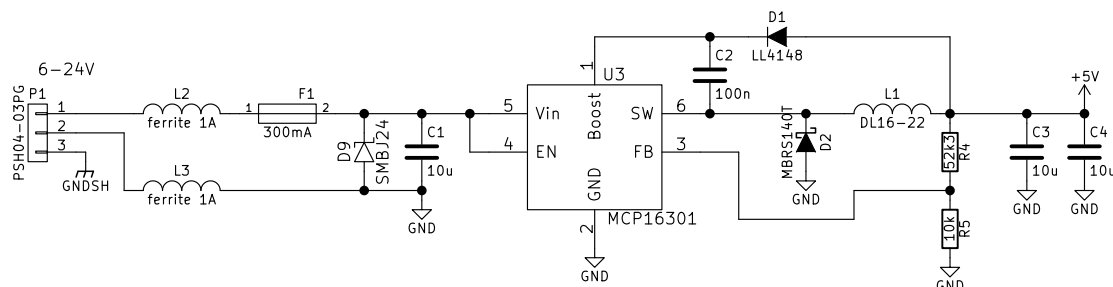
Stínění kabelů k motorům i napájení pro motory jsou navzájem spojena ale oddělena od ostatních stínění i od země na desce. Spojení stínění a GND desky bude provedeno přes kostru krabice nebo speciálním vodičem.



Obr. 5.4: Schéma zapojení motorových driverů

5.5 Napájecí obvod

Pro napájení byl vybrán obvod MCP16301. Je to integrovaný obvod určený pro snižující spínané zdroje. Maximální vstupní napětí je 30 V a maximální výstupní proud je 600 mA při 5 V. Zapojení je převzato z katalogu [17]. Feritky L2 a L3 slouží k odrušení. Výstupních 5 V je přivedeno na LDO regulátor na platformě Nucleo, jehož výstupem je 3,3 V.



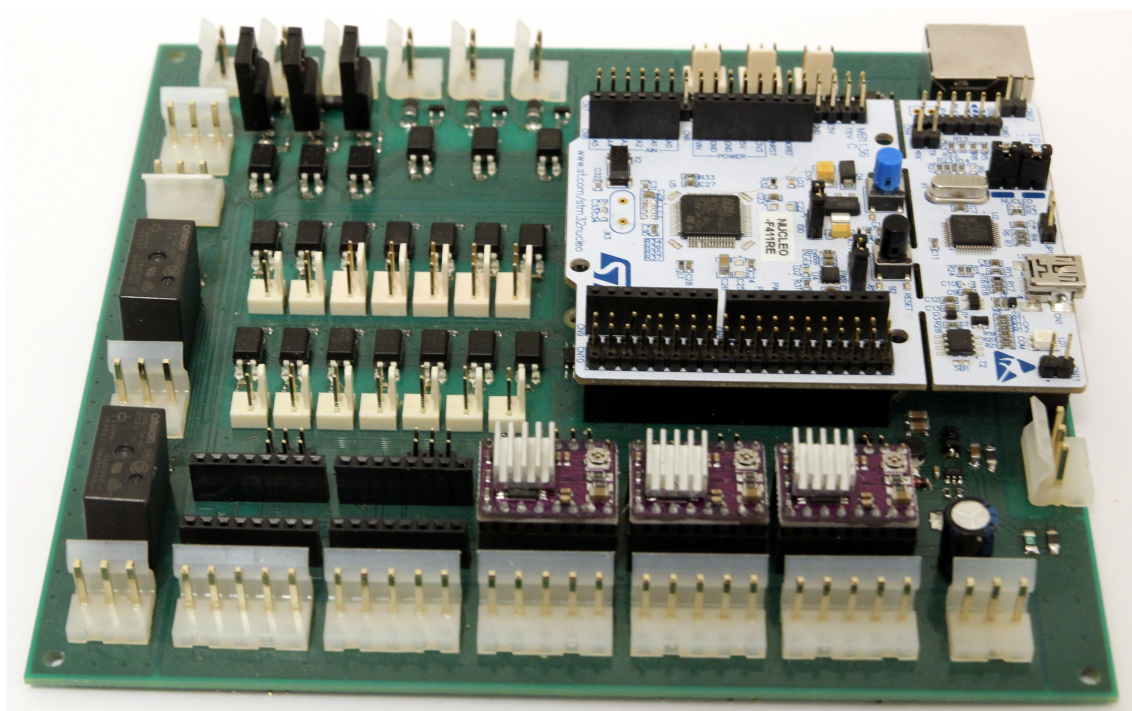
Obr. 5.5: Schéma zapojení napájecího zdroje

5.6 Návrh DPS

Deska plošných spojů byla navržena jako oboustranná se součástkami z jedné strany. Uprostřed desky je umístěn slot pro kit STM32f411-Nucleo.

Při návrhu bylo nutné počítat s velkou mírou rušení od spínaných zátěží jako jsou např. motory. Kvůli tomu je na DPS rozlita měď. Rozlita měď slouží pro rozvedení země.

U výkonových prvků bylo nutné počítat s velkými proudy, proto jsou konektory pro motory spojeny s drivery spojem o šířce 1,8 mm. Spoje pro připojení topných těles jsou stejně široké. Spoje široké 1,8 mm umožňují vést proud přibližně do 6 A.

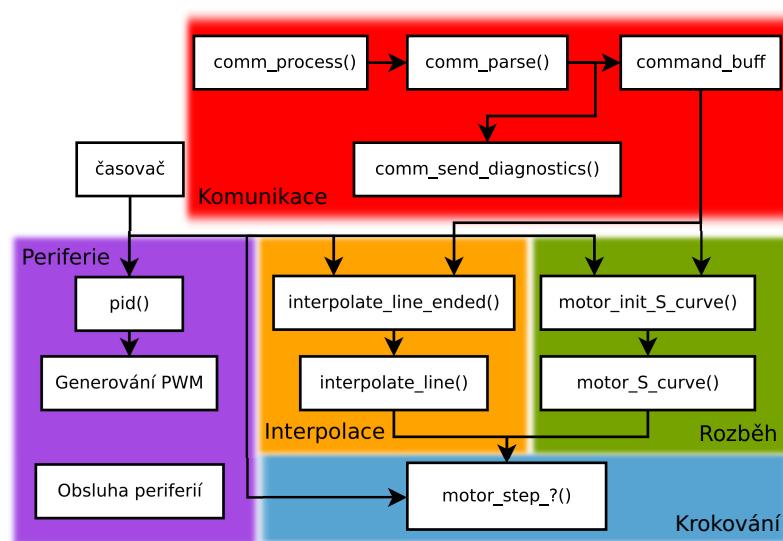


Obr. 5.6: Výsledná osazená DPS

6 STRUKTURA ŘÍDICÍHO FIRMWARE

Řídicí firmware musí zajišťovat kompletní řízení 3D tiskárny. V základu je firmware postaven na operačním systému reálného času FreeRTOS, který zajišťuje běžnou obsluhu periférií a také počítá časově náročné výpočty potřebné pro interpolaci. Pro časově kritické činnosti jako je vykonávání jednotlivých kroků, je vyhrazen časovač s přerušením. Přerušování se zpracovávají mimo RTOS, aby se zajistila dobrá doba odezvy.

Strukturu řídicího firmwaru je možné rozdělit na několik základních bloků, které jsou na sobě téměř nezávislé. Jednotlivé bloky, jejich propojení a funkce, které je tvoří, znázorňuje obrázek 6.



Obr. 6.1: Blokové schéma řídicího firmware

6.1 Blok komunikace s PC

Blok zajišťuje získávání příkazů z PC a současně odesílání stavových informací do PC. Stavovými informacemi je myšlena například aktuální pozice, teplota podložky, teplota extrudéru nebo aktuální rychlost.

6.1.1 Komunikační protokol

Komunikační protokol je navržen tak, aby bylo možné odhalit chyby v přenosu, které se mohou vyskytnout vlivem rušení. Protože je využita sběrnice RS-485, která je poloduplexní, je vhodné, aby na sběrnici byla právě jedna stanice v roli mastera a jedna nebo více stanic podřízených, které pouze odpovídají. Komunikační protokol

počítá pouze s jednou slave stanicí (3D tiskárnou). Roli mastera bude zastávat PC a podřízeným zařízením bude tiskárna.

Pole	Počet bytů	Hodnota	Význam
STX	1	0x02	byte uvozující rámeček
rezerva	1	0x00	volný byte použitelný např. pro číslování rámečků
typ dat	1	viz tab. 6.2	informace o typu dat
délka dat	1	0-255	velikost dat v bytech
data	0-255	viz tab. 6.2	vlastní data
CRC	2	-	kontrolní součet
ETX	-	-	ticho na sběrnici

Tab. 6.1: Struktura rámečku

Rámeček se skládá z jednoho uvozovacího bytu, který slouží k rozlišení začátku zprávy. Následuje jeden volný byte rezervovaný pro budoucí využití např. pro číslování rámečků nebo pro případnou adresaci zařízení na sběrnici. Následuje byte informující o typu rámečku. Typ určuje, jestli rámeček nese informaci o následující větě programu, nebo jestli se jedná o některý z řídicích rámečků. Úplný seznam typů rámečků je v tabulce 6.2.

Čtvrtý byte obsahuje informaci o délce pole data, které nese samotnou informaci. Pole data může svoji velikost měnit v rozsahu od 0 do 255 bytů. Na konci rámečku je umístěn 32-bitový kontrolní součet zajištěný cyklickým kódem CRC. Konec rámečku je ukončen klidem na sběrnici, jehož trvání musí být dostatečně dlouhé na to, aby mohla být zpráva zpracována.

Typ dat	Hodnota	Význam	Obsah pole data
GET	'?' = 0x3F	dotaz	prázdné
ACK	'A' = 0x41	potvrzení	
NACK	'N' = 0x4E	odmítnutí / chyba	
G-kód	'G' = 0x47	věta programu	celá věta programu v textové podobě bez komentářů
Stav	'S' = 0x53	informace o stavu stroje	viz tab. 6.3

Tab. 6.2: Typy dat

Stavové informace jsou přenášeny ve formátu JSON. Tento formát byl vybrán z důvodu, že je lidsky čitelný, na straně PC snadno parsovatelný a na straně tiskárny jej lze snadno vytvořit. Další výhodou tohoto formátu je, že strukturu posílaných

dat je možné velice efektivně měnit. Kompletní popis formátu přenášených stavových informací je v tabulce 6.3.

Pole	Typ	Význam
X	float	pozice osa X
Y		pozice osa Y
Z		pozice osa Z
T1		teplota č.1
T2		teplota č.2
T3		teplota č.3
sp		aktuální rychlost posuvu
buf	number	počet vět programu v bufferu

Tab. 6.3: Struktura stavových dat

6.1.2 Způsob komunikace

Komunikace po sběrnici je vždy zahajována ze strany PC. Počítač pravidelně posílá dotaz na stavové informace. Zjistí-li že v bufferu tiskárny je prostor pro další větu programu, může ji následně odeslat. Dotaz na stavové informace se provádí odesláním rámce s typem nastaveným na „?“ s nulovou délkou dat. Nová věta programu je zaslána do tiskárny rámcem s typem „G“ s daty obsahujícími větu programu. Programových vět je možné v jedné zprávě poslat i více. Jednotlivé věty musí být oddělené mezerou a nesmí obsahovat komentáře. Číslování vět v G-kódu se také nesmí posílat.

6.1.3 Implementace komunikace

Obsluha komunikace je obsažena v souborech *communication.h* a *communication.c*. Pro komunikaci byla využita periferie USART1 (univerzální jednotka sériové linky). Příjem a odesílání dat je zajištěno pomocí DMA a přerušování vyvolávaného tichem na sběrnici. Jednotka je neustále v režimu naslouchání a pouze pokud přijde požadavek na odeslání dat, dojde k přepnutí režimu na vysílání.

Vysílání probíhá tak, že se nejdříve nastaví směr pro budič linky RS485 na vysílání. Následně se nastaví DMA kanál tak, aby přenášel požadovaná data. Nakonec se přenos spustí. Po ukončení přenosu, který vyvolá přerušování, je směr opět přepnut na příjem.

Příjem dat je komplikovanější, protože není předem známo, jak velká bude přijímaná zpráva. Řešení problému se nabízí tři. První možnost je přijmout pouze

hlavičku, zjistit si jak velká data budou přenášena a poté přijmout zbytek. Toto řešení není možné použít, protože čas potřebný pro vyhodnocení délky dat a následné znovunastavení DMA kanálu by byl příliš dlouhý a docházelo by ke ztrátám dat v důsledku zastaveného přenosu.

Další možností by bylo přijímat do kruhového bufferu a vždy jednou za čas si nové zprávy vyčíst. Toto řešení by ovšem vyžadovalo komplikovanější protokol s přesně definovaným začátkem i koncem, aby bylo možné od sebe odlišit jednotlivé zprávy. To se obvykle dělá tak, že se vyhradí dva znaky, jeden pro začátek a druhý pro konec zprávy. Pokud se tyto znaky vyskytují v datech tak se zdvojují, aby bylo zřejmé, že nejde o řídicí znak. Problémy způsobené se zdvojováním některých znaků a následným oddělování zpráv tento způsob řešení vylučují.

Jak se ukázalo, ideálním řešením je nastavit DMA na příjem zprávy s maximální délkou a příjem ručně ukončit v okamžiku, kdy se detekuje ticho na sběrnici. Tento způsob příjmu se vyznačuje velice nízkými nároky na výpočetní výkon a vysokou spolehlivostí. Jelikož MCU STM32F4 mají možnost přerušení v případě ticha na sériových linkách, bylo toho využito.

V případě ukončeného příjmu je z přerušení zavolána funkce `comm_process()`, která zajistí přepnutí příjmových bufferů. Příjmové buffery jsou použity dva, a pravidelně se střídají. Následně se zkontroluje CRC. Souhlasí-li CRC vypočtené z přijaté zprávy a CRC přijaté v její hlavičce znamená to, že zpráva dorazila bez chyb a může být předána parseru.

Z přijaté zprávy jsou jednotlivé položky rozbaleny pomocí struktury `frame_t`. Struktura obsahuje hlavičku tak, jak je posílána. Pro zjištění položek potom stačí nakopírovat příchozí data byte po byte do vytvořené proměnné, která je typu `frame_t`. Kopírování se děje přímo při příjmu pomocí DMA.

6.2 Parser a interpret

Parser zajišťuje dekodování příchozích programových vět v G-kódu. Z G-kódu se dekodují informace o vzdálenostech, rychlosti, nastavených teplotách apod. Vyparsovaná data jsou předána interpretu, který určí jakým způsobem s nimi bude zacházeno a spustí příslušné akce.

Hlavní část parseru je ve funkci `void comm_parse(frame_t* frame)`. Parser je koncipován jako smyčka, která prochází znak po znaku celou větou. Při procházení textu parser postupně skládá čísla tak, že jakmile nalezne číslici, vynásobí dosavadní hodnotu parametru číslem 10 a přičte číslici. Pokud se narazí na písmeno uvozující další parametr, uloží se hodnota parametru, který právě skončil.

Desetinná čísla se parsují obdobně jako celá. Jediným rozdílem je, že pokud se při parsování narazí na oddělovač desetinných míst, parser si tuto skutečnost poznačí.

Při dalších nalezených číslicích násobí pomocnou proměnnou 10. Výslednou hodnotu v desetinném tvaru je poté možné získat vydělením vyparsovaného parametru pomocnou proměnnou.

Hodnoty koncových bodů jsou ukládány jako relativní pozice v krocích. Přepočet na kroky se provádí v celočíselné aritmetice. Je to zajištěno postupným výpočtem, kdy se nejprve vynásobí relativní vzdálenost s rozlišením, udávajícím počet kroků na jednotku vzdálenosti. Následně se převede hodnota, postupem popsáným výše, na desetinnou formu. Protože pohyby je možné vykonávat pouze v celých krocích, desetinná část se neuvažuje.

Hodnoty jednotlivých parametrů, vyparsovaných z G-kódu, se ukládají do struktury `command_t`. Výsledná struktura se po naplnění ukládá do bufferu, ze kterého je vyčítána při interpolaci.

6.3 Interpolace

Interpolátor vypočítává, na kterém motoru je třeba vykonat krok a kterým směrem. Využívá k tomu interpolační algoritmus DDA popsáný v kapitole 4.

Vlastní interpolátor je situován do dvou funkcí. Funkce sloužící k inicializaci interpolátoru se nazývá `init_interpolation_line()`. Zajišťuje výchozí nastavení proměnných, které jsou nezbytné pro vlastní výpočet interpolace. Dále také rozhoduje o směru pohybu, podle parametrů vyparsovaných z G-kódu. Druhá funkce, `interpolate_line()`, slouží pro vlastní interpolaci. Je volána pravidelně v přerušování. Výstupem každého volání je směr pohybu v každé ose a informace o tom, ve kterých osách se má krok vykonat a ve kterých ne. Výstupní informace je typu `step_t`, má podobu bitového pole o celkové velikosti 1 byte. Položky `dir_<x,y,z>` popisují směr pohybu v dané ose, položky `step_<x,y,z>` určují jestli se krok provede nebo ne.

6.4 Krokování

Vykonávání pohybů na základě informací získaných z interpolátoru probíhá v přerušování, vyvolávaném časovačem. Perioda přerušování určuje rychlost pohybu. Firmware podporuje dva typy kinematiky. Základní XY kinematiku a kinematiku H-bot. Přepínání typů kinematiky je zajištěno podmíněným překladem. Jeli makro `H_KINEMATICS` přítomno při překladu, je vybrána H-bot kinematika, chybí-li makro bude se překládat XY kinematika.

Výchozí je kinematika XY. Při použití H-bot jsou základní pohyby převáděny pomocí několika podmínek, které v závislosti na směru pohybu, určují kterými motory

a kterým směrem se bude otáčet.

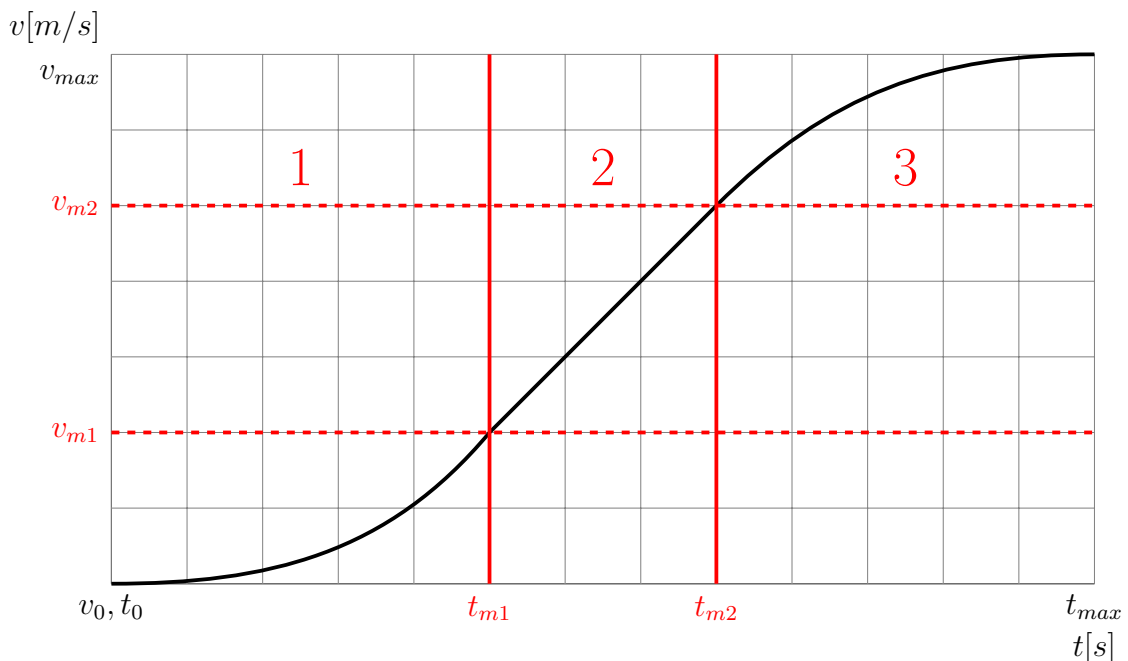
Krokovací funkce zajišťují vykonání kroku. Modifikují také interní proměnnou indikující aktuální pozici.

6.5 Rozběhový profil

Ve firmwaru je implementován rozběhový profil po S-křivce. Byl vybrán na základě výborné schopnosti zamezit ztrátám kroku a také kvůli plynulosti pohybu.

Výpočet rychlosti se provádí pravidelně v přerušení časovače. Výstupem je rychlost udávaná za pomoci hodnoty děličky kmitočtu. Základním kmitočtem je volána funkce pro výpočet rychlosti. Aktuální rychlost je dána hodnotou děličky základního kmitočtu. Dělička je realizována pomocí inkrementované proměnné, která po dosažení hodnoty vypočtené z rychlosti vyvolá rutinu pro vykonání pohybu. Z toho plyne, že výpočet rozběhového profilu se vykonává častěji než samotné krokování.

Kmitočet časovače lze odvodit z maximální rychlosti vyjádřené jako počet kroků za 1 s. Toto vyjádření přímo odpovídá minimální frekvenci základního kmitočtu. Protože se dělička chová nelineárně v oblasti nízkých hodnot, je vhodné ještě základní kmitočet zvýšit. Se zvyšujícím se kmitočtem roste rozlišení s jakým je možné nastavovat rychlost.



Obr. 6.2: Detail rozběhu

Samotný výpočet je situován do funkce `motor_S_curve()`. Výpočet S-křivky se provádí po částech pomocí stavového automatu. Průběh křivky spolu s vyznačenými

oblastmi výpočtu je zobrazen na obrázku 6.2.

Oblast č. 1 je charakteristická lineárně narůstajícím zrychlením. Je ukončena v okamžiku, kdy rychlost vzroste nad polovinu požadované rychlosti, nebo když je překročeno maximální povolené zrychlení.

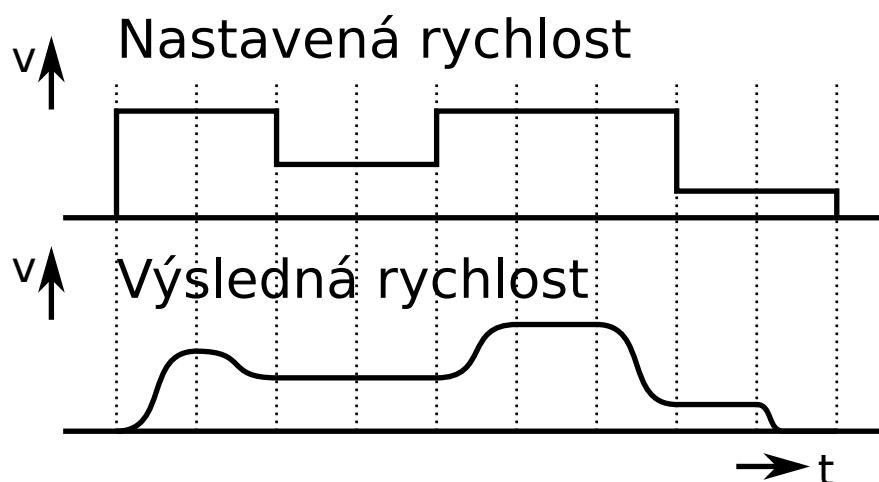
Výpočet následně pokračuje buď lineární částí (oblast č. 2), kdy se zrychlení nemění, nebo částí kde, zrychlení lineárně klesá (oblast č. 3).

V oblasti č. 2 se dosažené zrychlení nemění, ale je neustále přičítáno k rychlosti. Část je ukončena v okamžiku, kdy je nutné začít snižovat zrychlení tak, aby bylo nulové právě v okamžiku dosažení požadované rychlosti. Mez v_{m2} lze vypočítat ze vztahu:

$$v_{m2} = v_{max} - (v_{m1} - v_{m0}) \quad (6.1)$$

V oblasti č. 3 se postupně snižuje hodnota zrychlení, dokud není nulové. V okamžiku nulového zrychlení výpočet končí a nadále je rychlost udržována jako konstantní. Pro zpomalování platí stejná pravidla jako pro zrychlování jen inverzně.

Při reálném použití jsou jednotlivé pohyby skládány za sebe s různými rychlostmi. Přechody mezi jednotlivými rychlostmi jsou asi nejproblematictější částí implementace rozběhového profilu po S-křivce. Komplikace spočívá v tom, že je nutné přesně znát čas, kdy dojde k ukončení interpolace s danou rychlostí. Ovšem známé je pouze to, kolik kroků zbývá vykonat. Protože intervaly mezi jednotlivými kroky nejsou konstantní, ale mění se spolu s rychlostí, je výpočet potřebného času značně problematický.



Obr. 6.3: Průběh rychlostí S-křivky

Výše popsany problém s výpočtem času, je hlavním důvodem, proč se rozběhový profil po S-křivce využívá zejména v profesionálních strojích. Implementovaný

rozběhový profil je generován pomocí předpočítané tabulky. Tabulka obsahuje informace o tom, jak dlouho trvají jednotlivé části S-křivky. V tabulce jsou také hodnoty rychlostí, kdy dojde ke zlomům. Při samotném generování S-křivky je sledována aktuální rychlost a podle ní jsou měněny parametry generování.

6.6 Obsluha periférií

K tiskárně je připojeno množství periférií, které jsou nezbytné pro její funkci. Mezi periferie lze zařadit snímače výchozí polohy, bezpečnostní spínač nebo měření teplot extrudérů a podložky. Obsluha periférií probíhá periodicky ve vlákne RTOS. Perioda obsluhy je nastavena na 100 ms.

Všechny vstupní periferie s binárním výstupem, kromě bezpečnostního spínače, jsou připojeny přes expandér. Expandér je složený z posuvného registru. Vyčítání binárních vstupů probíhá ve smyčce, kdy se postupně vyčítají hodnoty jednotlivých vstupů posuvného registru. Hodinová frekvence vyčítání registru je přibližně 1,5 MHz.

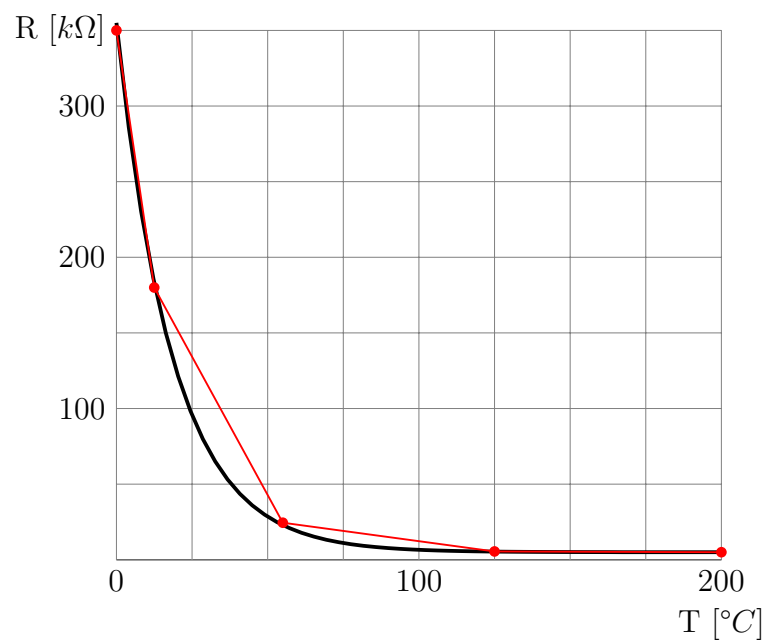
Vyčtené hodnoty se ukládají do typu nadefinovaného jako unie s 16-ti bitovým číslem a bitovým polem. Pomocí 16-ti bitového čísla se ukládají hodnoty. Pro snadný přístup k vyčteným hodnotám je určeno bitové pole.

Měření teploty se provádí pomocí interního analogově-digitálního převodníku. Využity jsou 3 kanály jednoho převodníku. Vyčítání se provádí automaticky. Je vyhrazen prostor, kam DMA cyklicky zapisuje naměřené hodnoty. Měření se tedy provádí neustále bez účasti CPU. S periodou 100 ms se naměřené hodnoty kopírují do pole, které slouží pro výpočet plovoucího průměru. Průměrná hodnota se vypočítává těsně před tím, než se spustí výpočet regulátorů.

Protože jako snímače teploty jsou použity termistory, je měřené napětí nelineární. Pro přepočtení naměřeného napětí na teplotu se používá tabulka. Tabulka určuje jaké napětí odpovídá jaké teplotě. V tabulce je několik záznamů, které rozdělují měřený rozsah na několik úseků. Při přepočtu se v tabulce vyhledají příslušné meze daného úseku. Ze získaných mezí a naměřené hodnoty se lineární interpolací vypočítá hodnota teploty.

Na obrázku 6.4 je vidět typická charakteristika termistoru (černě) a interpolace pomocí tabulky s 5 body v měřeném rozsahu. Čím větší bude počet interpolačních bodů, tím bude výsledek přesnější. S výhodou je možné umístit více bodů do oblasti ohybu charakteristiky a méně bodů do lineární části.

Samotný výpočet se provádí s čísly ve formátu plovoucí desetinné čárky. Aby se snížila výpočetní náročnost, jsou v interpolační tabulce uvedeny hodnoty napětí v podobě v jaké jsou získávány z ADC. Odpadá tak nutnost provádět přepočtení na



Obr. 6.4: Charakteristika termistoru s naznačenou interpolací

reálné napětí. Interpolace bodů se provádí podle obecného vzorce:

$$y = y_0 + (x - x_0) \frac{y_1 - y_0}{x_1 - x_0} \quad (6.2)$$

kde:

- y_0 hodnota napětí dolní meze oblasti
- y_1 hodnota napětí horní meze oblasti
- x_0 hodnota teploty dolní meze oblasti
- x_1 hodnota teploty horní meze oblasti
- x naměřená hodnota napětí

7 PSD REGULÁTORY

Nezbytnou součástí každé FDM tiskárny je i regulace teplot extrudérů a tiskové podložky. K tomu slouží PSD regulátory, které pomocí PWM spínají výkonové tranzistory. PSD regulátor lze popsat rovnicí [2]

$$u(t) = K_R[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt}] \quad (7.1)$$

Protože není možné efektivně počítat derivaci, aproximuje se výpočtem difference neboli rozdílu. Integrace se aproximuje jako suma.

Při použití aproximací vyplývá rovnice:

$$u(k) = K_R[e(k) + \frac{T}{T_I} \sum_{i=0}^{k-1} e(i) + \frac{T_D}{T} (e(k) - e(k-1))] \quad (7.2)$$

Z poslední rovnice vychází i celý algoritmus výpočtu, kdy se nejprve získá hodnota regulační odchylky e_t vypočtením rozdílu mezi požadovanou hodnotou a hodnotou právě změřenou na výstupu systému. Dále se vypočítává sumační složka, ta je dána předchozí hodnotou sumace a regulační odchylkou. Nakonec se vypočte hodnota difference z předchozí hodnoty odchylky a aktuální hodnoty odchylky. Hodnota akčního zásahu je dána součtem jednotlivých složek, které jsou vynásobeny příslušnými koeficienty, které určují chování regulátoru.

U diskrétních regulátorů se výrazně projevuje tzv. wind-up jev. Kdy dochází k přeintegrování. V tomto případě může regulátoru dlouho trvat, než zareaguje na změnu požadované hodnoty. Tento problém je možné řešit omezením maximální hodnoty integrace. Obvykle se volí omezení na maximální hodnotu výstupní veličiny.

U spojitých regulátorů tento jev obvykle nepozorujeme, neboť hodnota integrace bývá přirozeně omezena napájecím napětím. Zatímco diskrétní regulátor může integrovat tak dlouho než dojde k přetečení.

Pro správnou činnost PSD regulátorů je třeba určit jejich parametry v závislosti na regulované soustavě. Protože se tisková podložka a extrudér chovají odlišně bylo nutné navrhnout dva rozdílné regulátory.

7.1 Návrh regulátoru pro extrudér

Před návrhem regulátoru je nejprve nutné zjistit chování regulované soustavy. Pro extrudér je nejvhodnější měřit odezvu na jednotkový skok, protože je soustava pomalá.

Při měření byla nastavena střída PWM signálu na pevnou hodnotu a měřila se teplota s periodou 500 ms. Měření probíhalo dokud neodezněl přechodný děj.

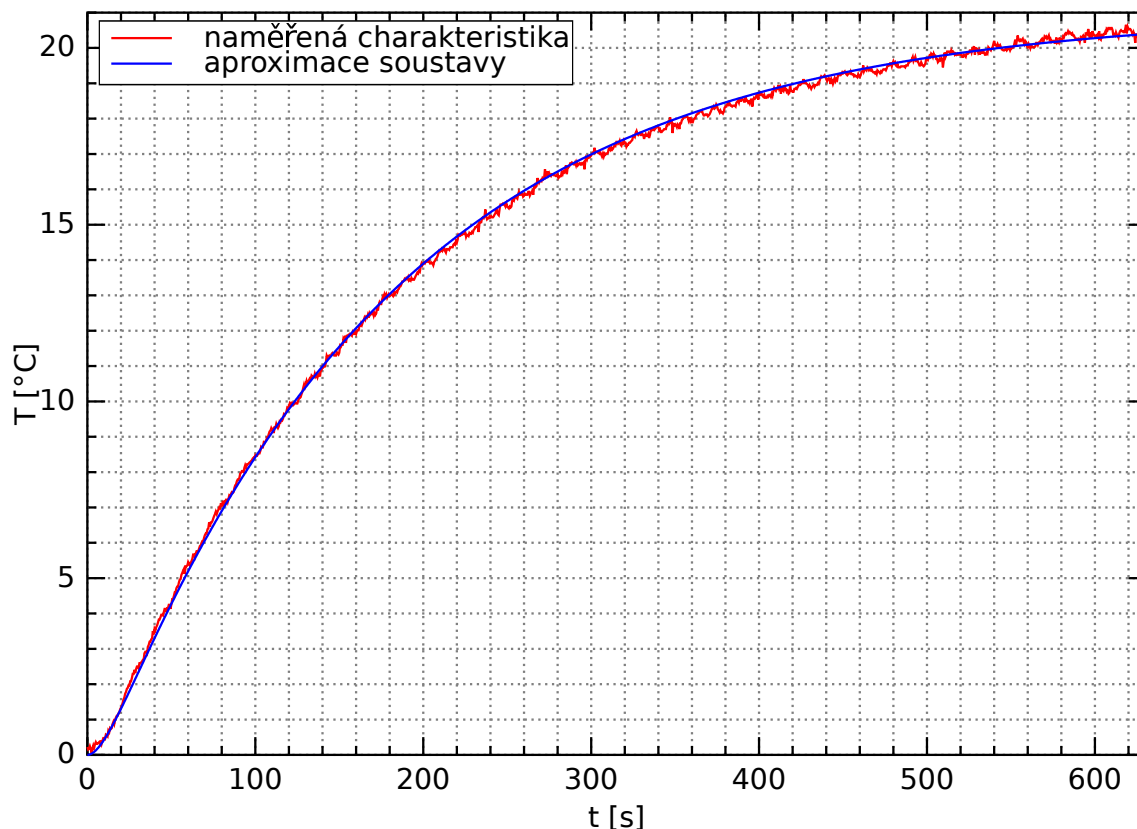
Výslednou charakteristiku bylo poté nutné upravit tak, aby počáteční teplota byla nulová tzv. normalizace. To bylo provedeno odečtením počáteční teploty.

Následně bylo nutné určit aproximaci soustavy. Protože na začátku je patrný průtah charakteristiky, bylo vhodné aproximovat soustavou druhého řádu. Časové konstanty byly určeny z přechodové charakteristiky. Vede-li se tečna v inflexním bodě, je možné, v místě protnutí nulové teploty, odečíst hodnotu první časové konstanty. Pro extrudér je první časová konstanta 10 s. Druhá časová konstanta odpovídá bodu, kdy tečna dosáhne maximální teploty. Byla určena na 175 s. Zesílení soustavy je 21, je dáno výškou změřené charakteristiky.

Ze zjištěných časových konstant je možné určit přenos.

$$F_s = \frac{K}{(T_1p + 1)(T_2p + 1)} = \frac{21}{(10p + 1)(175p + 1)} \quad (7.3)$$

Pro kontrolu byly zobrazeny obě charakteristiky v jednom grafu. Na obrázku 7.1 je vidět, že aproximace přesně odpovídá naměřené charakteristice.



Obr. 7.1: Naměřená a aproximovaná normovaná přechodová charakteristika extrudéru

Pro prvotní návrh se nejvíce hodí metoda Ziegler-Nicholse. Je založená na zjištění kritických parametrů K_{krit} T_{krit} a následném určení vhodných konstant pomocí

tabulky. Je-li známa přechodová charakteristika, je možné přibližné hodnoty kritických parametrů získat přímo z doby náběhu a průtahu.

Pro určení kritického zesílení, byl zvolen výpočet kritických parametrů pomocí přechodové charakteristiky. Z charakteristiky se zjistilo, že $T_u = 10\text{ s}$ a $T_n = 175\text{ s}$. Pro kritické hodnoty platí [2]:

$$K_{krit} \doteq \frac{\pi T_n}{2 T_u} + 1 = \frac{\pi 175}{2 \cdot 10} + 1 = 28,49 \quad (7.4)$$

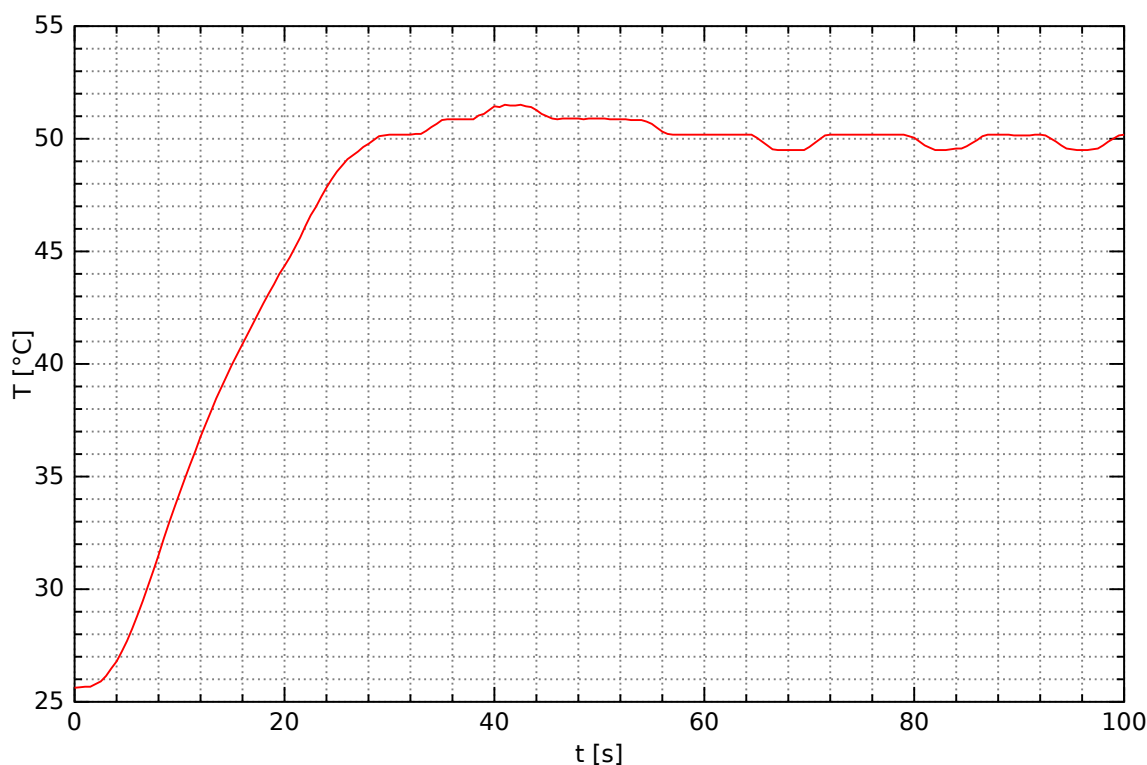
$$T_{krit} \doteq 4T_u = 4 \cdot 10 = 40\text{ s} \quad (7.5)$$

Pro jednotlivé parametry PSD regulátoru platí [2]:

$$K_R = 0,6K_{krit} = 0,6 \cdot 28,49 = 17,1 \quad (7.6)$$

$$T_I = 0,5T_{krit} = 0,5 \cdot 40 = 20\text{ s} \quad (7.7)$$

$$T_D = 0,125T_{krit} = 0,125 \cdot 40 = 5\text{ s} \quad (7.8)$$



Obr. 7.2: Přechodová charakteristika řízeného extrudéru

Výsledná přechodová charakteristika, kdy byla teplota regulována výše navrženým PSD regulátorem se vzorkovací periodou 500 ms, je vidět na obrázku 7.2.

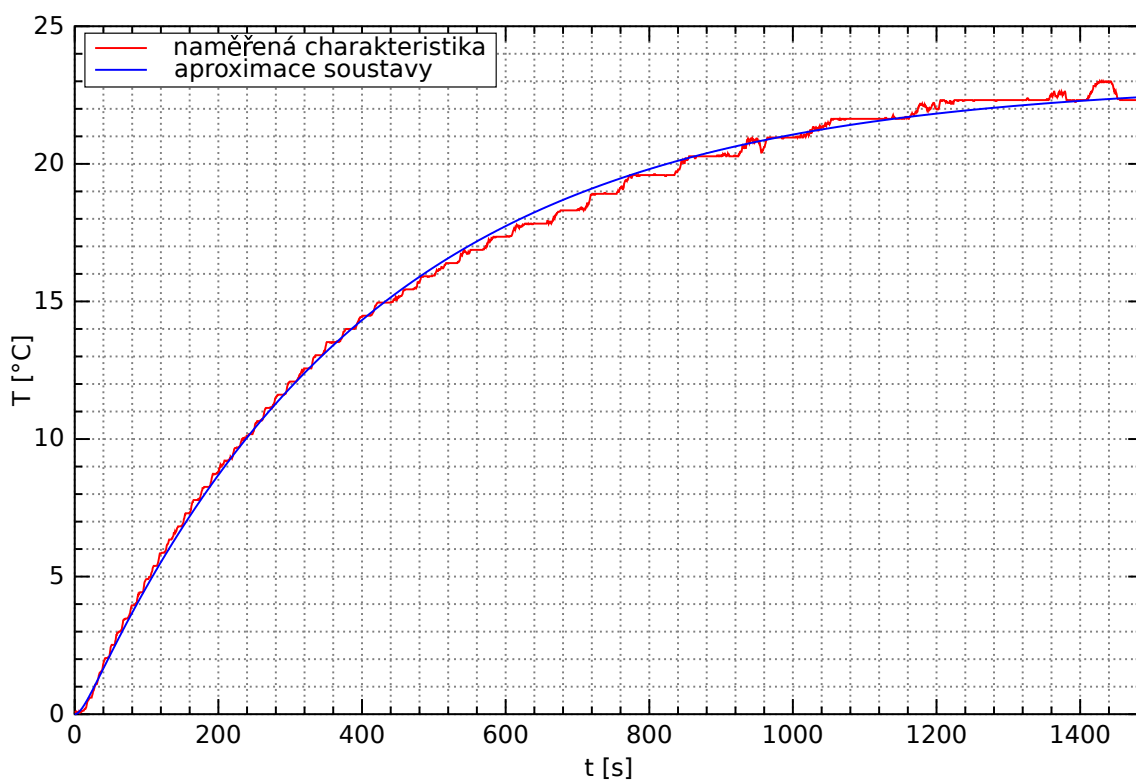
Požadovaná teplota byla nastavena na hodnotu 50 °C. Doba ustálení byla přibližně 60 s s překmitem 0,75 %.

7.2 Návrh regulátoru pro tiskovou podložku

Regulátor pro tiskovou podložku byl navržen stejným postupem jako regulátor pro extrudér. Nejprve se určila aproximace soustavy. Výsledný přenos aproximace je:

$$F_s = \frac{23}{(10p + 1)(400p + 1)} \quad (7.9)$$

Že je aproximace vhodná, lze vyvodit z obrázku 7.3. Charakteristiky se překrývají po celou dobu trvání přechodného děje.



Obr. 7.3: Naměřená a aproximovaná normovaná přechodová charakteristika tiskové podložky

Z přenosu vyplývá $T_u = 10 \text{ s}$ a $T_n = 400 \text{ s}$. V tomto případě je kritické zesílení $K_{krit} = 63,8$ a kritická perioda je $T_{krit} = 40 \text{ s}$.

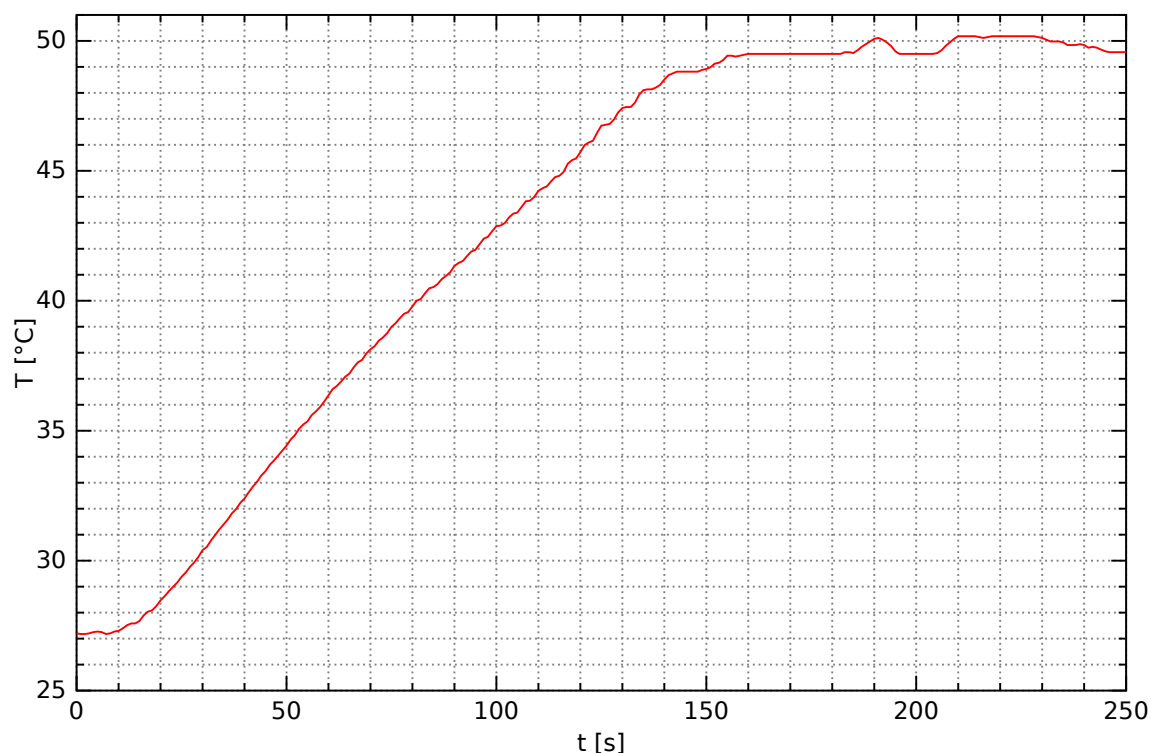
Pro jednotlivé parametry PSD regulátoru poté platí [2]:

$$K_R = 0,6K_{krit} = 0,6 \cdot 63,8 = 38,3 \quad (7.10)$$

$$T_I = 0,5T_{krit} = 0,5 \cdot 40 = 20 \text{ s} \quad (7.11)$$

$$T_D = 0,125T_{krit} = 0,125 \cdot 40 = 5 \text{ s} \quad (7.12)$$

Překmit regulace s použitím výše navrženého regulátoru byl přibližně 7 %. Zvýšením integrační časové konstanty na hodnotu 60 s, překmit klesl na 1,5 %. Nastavení T_I na hodnotu 75 s vede k regulaci zcela bez překmitu. Protože je soustava velmi pomalá bylo vhodné použít delší periodu vzorkování. Vzorkování se provádělo s periodou 1 s. Výsledná přechodová charakteristika je na obrázku 7.4.



Obr. 7.4: Přechodová charakteristika řízené tiskové podložky

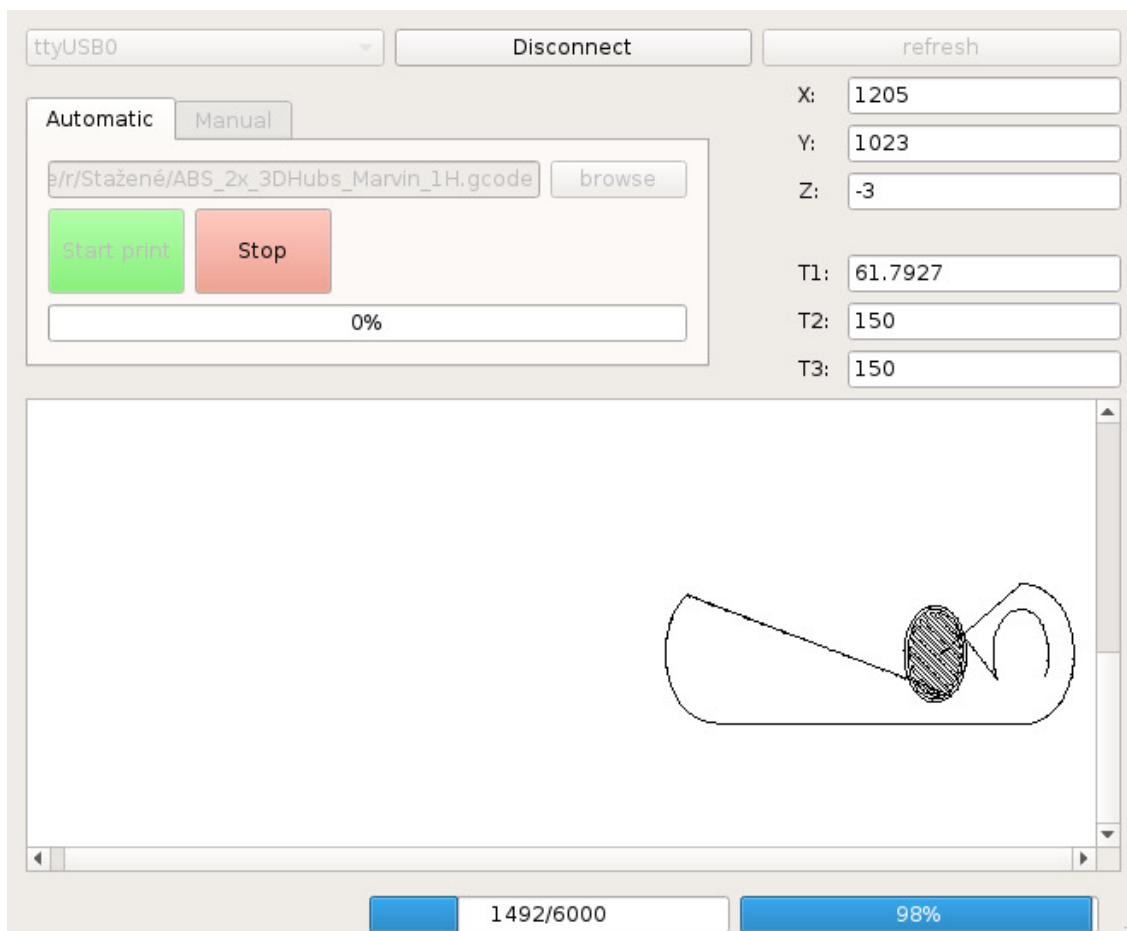
7.3 Shrnutí

Při měření řízeného systému, se jako největší problém jevila kvalita měření teploty. Drobné kolísání teploty způsobovalo velké překmity a dlouhou dobu ustálení. To se podařilo částečně vyřešit průměrováním měřených dat.

Při návrhu regulátorů se potvrdilo, že metoda Ziegler-Nicholse slouží spíše pro prvotní přibližné určení parametrů regulátorů. Pro jednoduché regulátory se jeví jako nejlepší, doladovat parametry experimentálně.

8 PC APLIKACE

Aby bylo možné s tiskárnou komunikovat, byla vytvořena aplikace pro PC. Aplikace umožňuje zobrazovat stavová data získávaná z 3D tiskárny, ručně ovládat pohyby a posílat tiskárně jednotlivé věty G-kódu načtené ze souboru.



Obr. 8.1: Vzhled vytvořené ovládací aplikace

8.1 Popis ovládací aplikace

Ovládací aplikace v horní části okna obsahuje panel pro připojení k tiskárně. Pomocí rolovací nabídky je možné vybrat sériový port, kam je tiskárna připojena. Tlačítkem *Connect/Disconnect* je možné se k tiskárně připojit, nebo se naopak odpojit. Tlačítko *Refresh* slouží k znovunačtení seznamu dostupných portů.

Uprostřed okna vlevo jsou dvě záložky. Každá záložka reprezentuje jeden režim činnosti. Záložka *Automatic* umožňuje načíst soubor s G-kódy a následně spustit automatický tisk. Program začne automaticky posílat jednotlivé věty programu do tiskárny. Záložka *Manual* umožňuje ruční ovládání pohybu tiskové hlavy.

V pravé části okna se zobrazují stavové informace. První 3 řádky zobrazují aktuální pozici. Zbylé řádky zobrazují teploty jednotlivých periférií.

Hlavní částí okna je plocha, kde se vykresluje aktuální trajektorie. K vykreslování dochází po jednotlivých vrstvách. Není zde rozlišeno, jestli se skutečně tiskne, nebo jde jen o přemístění hlavy bez vytlačování materiálu.

Ve stavovém řádku jsou dva prvky, indikující aktuální rychlost pohybu a zaplnění vnitřního bufferu.

8.2 Implementace PC aplikace

Aplikace je napsána v jazyku C++ s frameworkem Qt. Tato kombinace byla zvolena, aby bylo možné aplikaci přeložit pro libovolnou platformu. Aplikace je strukturovaná do třech tříd.

Třída `MainWindow` zajišťuje vykreslování okna, správu tlačítek a propojení všech součástí aplikace. Je zde nadefinováno chování všech widgetů.

Třída `comm` slouží pro komunikaci. Zapouzdřuje základní funkčnost knihovny `QtSerialPort`. Obsahuje slot `readInputData()`, který je napojen na příchozí data. Třída také kontroluje validitu příchozí zprávy. Kontroluje se CRC a dále platnost příkazu. Nakonec se kontroluje jestli příchozí věta G-kódu neobsahuje nedovolené parametry.

Je-li zpráva validní, je vyvolán signál `process(char cmd, QByteArray data)`. Signál je v hlavní třídě napojen na slot zpracování příchozích zpráv. Dále se zde nacházejí metody pro otevření a zavření portu nebo pro odeslání dat. Protože je možné ovládací aplikaci zkompilovat pro různé stroje, byly přidány funkce zjišťující endianitu.

Třída `gcodeFile` je určena pro zpracování souboru s G-kódem. Třída umožňuje načíst soubor a postupně na dotaz vrátit balík zpráv, které se mají poslat do tiskárny. Třída si také udržuje pozici v souboru a celkový počet programových vět. Z těchto informací je možné určit, jaká procentuální část byla odeslána do tiskárny.

Funkce `get_last_code()`; umožňuje získat naposledy odeslaný balík vět programu. To je důležité v případě, kdy dojde ke ztrátě posledního příkazu. Dojde-li ke ztrátě zprávy, je možné ji poslat znovu.

8.3 Činnost aplikace

Po připojení k tiskárně, po stisknutí tlačítka *Connect*, se spustí komunikace. Aplikace každých 10 ms posílá tiskárně dotaz na stav. Tiskárna odpovídá zprávou o stavu, která obsahuje pozici, zaplněnost bufferu atd.

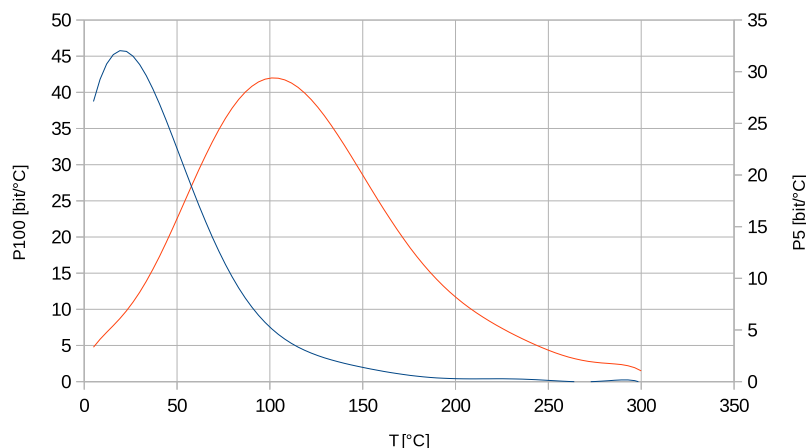
Pokud aplikace na základě stavových informací zjistí, že je místo v bufferu tiskárny a je spuštěn tisk ze souboru, odešle nové věty tiskového programu. G-kód se posílá po skupinách vět. Každá skupina obsahuje 4 věty. Tento způsob posílání G-kódu je důležitý, aby tiskárna měla vždy dostatek vět ke zpracování.

Tiskárna každou přijatou zprávu potvrzuje odesláním ACK zprávy. Pokud aplikace nedostane potvrzovací zprávu, automaticky odešle předchozí skupinu vět.

Manuální režim ovládání tiskárny je koncipován tak, že při stisku tlačítka pohybu dojde k odeslání příkazu G0 s patřičnými parametry. Pomocí rozbalovacího menu je možné vybrat vzdálenost pohybu.

9 ZHODNOCENÍ FUNKČNOSTI

Navržená DPS byla testována po jednotlivých částech. Při testování bylo zjištěno, že měření teploty má malé rozlišení. Týká se to zejména vyšších teplot. Je to způsobeno nevhodnou hodnotou rezistoru, který je připojen k termistoru a spolu s ním tvoří napěťový dělič. Možné řešení by bylo použít menší hodnotu rezistoru, a tím zvýšit rozlišení při vyšších teplotách. Další možnost je linearizovat průběh termistoru speciálními obvody k tomu určenými.



Obr. 9.1: Rozlišení měření teploty pro rezistor o hodnotě $100k\Omega$ (modrá) a $5k\Omega$ (oranžová)

Na obrázku je znázorněno teoretické rozložení rozlišení měření teploty. Při původně navržené hodnotě rezistoru $100k\Omega$ je rozlišení při teplotě 200°C přibližně $0,5 \text{ bit } ^\circ\text{C}^{-1}$. Pokud by se použil rezistor o hodnotě $5k\Omega$, bylo by rozlišení při teplotě 200°C rovno $8 \text{ bit } ^\circ\text{C}^{-1}$.

Navržený způsob komunikace fungoval naprosto spolehlivě. Ve výjimečných případech, kdy došlo k nedoručení zprávy, byla ztracená zpráva nahrazena.

Výsledná křivka rozběhového profilu má správný tvar, ale nejeví se jako zcela ideální. Například pokud má tiskárna pouze jeden příkaz, nedojde ke zrychlování. Je to způsobeno tím, že během jedné věty nemůže být rychlost zvyšována i snižována. Možným řešením by bylo počítat aproximaci rychlosti a časů s předstihem. To by mohlo vést k přibližnému zjištění bodů zlomu S-křivky. Aby bylo na tiskárně možné reálně tisknout, je nutné správně nastavit rozlišení jednotlivých os a zkalibrovat rychlost pohybů.

10 ZÁVĚR

V teoretické části práce byly popsány 3D tiskárny. Byly shrnuty základní metody 3D tisku a jejich vlastnosti. Dále se práce zabývala krokovými motory, zejména jejich principy a způsoby řízení. Zvláštní pozornost byla věnována rozběhovým profilům, zejména vlastnostem a způsobům implementace. Dále se práce zabývala interpolačními algoritmy. Byly popsány základní typy algoritmů používaných v CNC strojích.

V praktické části práce byla navržena řídicí deska. Deska umožňuje připojit dva extrudéry, tiskovou podložku, snímače pozice a několik dalších periférií potřebných pro tisk. Funkčnost výsledné desky byla ověřena na modelu 3D tiskárny.

Dále byl vytvořen firmware, který celou tiskárnu řídí. Do řídicího firmwaru byl implementován rozběhový profil po S-křivce, který zajišťuje hladký průběh pohybu. Rozběhový profil po S-křivce byl vybrán pro jeho schopnost zamezit ztrátě kroku. Firmware dále obsahuje interpolační algoritmus. Pro implementaci byl vybrán lineární algoritmus DDA. Výsledný firmware umožňuje řídit tiskárnu pomocí vybraných příkazů G-kódu. Jednotlivé příkazy jsou do tiskárny nahrávány z PC pomocí vytvořené aplikace. Součástí řídicího firmwaru je také regulace teplot tiskové podložky a extrudéru. Regulaci teplot zajišťují dva regulátory typu PSD. Návrh parametrů regulátorů byl proveden na základě naměřených průběhů teploty. Při návrhu PSD regulátorů byla využita metoda Ziegler-Nicholse.

Vytvořená PC aplikace umožňuje ručně ovládat 3D tiskárnu pomocí tlačítek. Aplikace také umožňuje posílat tiskárně příkazy uložené v souboru. Tiskárna v tomto případě automaticky vykonává zadaný program. Aplikace také obsahuje informace o stavu tiskárny, kde je možné zjistit aktuální pozici a hodnoty teplot extrudéru a tiskové podložky.

Výsledkem práce je řídicí deska pro 3D tiskárnu, která je řízena pomocí MCU. Realizovaná deska umožňuje pohyb extrudéru po zadané trajektorii. Dále je schopná regulovat teploty extrudéru a podložky na určenou hodnotu. Vše je ovládáno z PC pomocí vytvořené aplikace.

LITERATURA

- [1] AREVALO, Victor M.; OPTICOS, A.; ENSENADA, B. C. Sinusoidal Velocity Profiles for Motion Control. *Creport. Augen Opticos, SA, México*.
- [2] BLAHA, Petr a Petr VAVŘÍN. *Řízení a regulace 1* [online]. V 1.3.5. Brno: VUT Brno, 2005 [cit. 2017-05-23]. Dostupné z: <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbmXyaXplbmhcmVndWxhY2UxfGd4OjYxNTNlYmMONTJjYmViZTE>
- [3] JOY, Kenneth I. BRESHENHAM'S ALGORITHM. *On-Line Computer Graphics Notes* [online]. Davis: University of California, 1999, , 15 [cit. 2017-05-24]. Dostupné z: <http://graphics.idav.ucdavis.edu/education/GraphicsNotes/Bresenhams-Algorithm.pdf>
- [4] KOREN, Y. a O. MASORY. Reference-Pulse Circular Interpolators for CNC Systems. *Journal of Engineering for Industry* [online]. 1981, **103**(1), 131- [cit. 2016-12-19]. DOI: 10.1115/1.3184454. ISSN 00220817. Dostupné z: <http://ManufacturingScience.asmedigitalcollection.asme.org/article.aspx?articleid=1446242>
- [5] LEWIN, Chuck. S-Curve Profiles Deep Dive. In: *Performance Motion Devices* [online]. [cit. 2016-11-06]. Dostupné z: <https://www.pmdcorp.com/resources/articles/18>
- [6] LINKEOVÁ, Ivana. Interpolární NURBS křivky. In: *Ivana Linkeová* [online]. [cit. 2016-12-26]. Dostupné z: http://marian.fsik.cvut.cz/~linkeova/vyuka/pubic/novemesto2006_li.pdf
- [7] NGUYEN, Kim Doang, Teck-Chew NG a I-Ming CHEN. On Algorithms for Planning S-curve Motion Profiles. *International Journal of Advanced Robotic System*. 2008, **5**(1), 8.
- [8] SASCHA, Weikert a et al. Modeling and measurement of h-bot kinematic systems. *American Society for Precision Engineering*. Denver, 2011.
- [9] SUH, Suk-Hwan. *Theory and design of CNC systems*. 1. London: Springer, c2008. ISBN 9781848003361.
- [10] *Design Trends* [online]. 2010, **2010**(3) [cit. 2016-11-06]. Dostupné z: http://www.motion-designs.com/images/DTrends_Aug_2010.pdf
- [11] Hybrid Stepper Motor. *Circuit globe* [online]. [cit. 2016-12-23]. Dostupné z: <http://circuitglobe.com/hybrid-stepper-motor.html>

- [12] Permanent Magnet Stepper Motor. *Circuit globe* [online]. [cit. 2017-05-24]. Dostupné z: <http://circuitglobe.com/permanent-magnet-stepper-motor.html>
- [13] *Prusa Research* [online]. Praha, c2017 [cit. 2017-05-24]. Dostupné z: <http://www.prusa3d.cz/>
- [14] Krokový motor. *Pohonnatechnika.cz* [online]. [cit. 2016-12-23]. Dostupné z: <http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>
- [15] Krokový motor - princip. *Mylms* [online]. [cit. 2017-05-24]. Dostupné z: <https://www.mylms.cz/text-krokovy-motor-princip/>
- [16] *Formlabs* [online]. c2017 [cit. 2017-05-24]. Dostupné z: <https://formlabs.com/>
- [17] MCP16301: High-Voltage Input Integrated Switch Step-Down Regulator. *Microchip* [online]. Microchip, 2015 [cit. 2016-12-27]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/20005004D.pdf>
- [18] Motion Control Overview. *ADVANCED Motion Controls* [online]. c2017 [cit. 2017-05-25]. Dostupné z: <https://www.a-m-c.com/experience/technologies/motion-control/overview/>
- [19] *3D-tisk.cz* [online]. Brno: Vydavatelství Nová média, c2014 [cit. 2017-05-24]. Dostupné z: <http://www.3d-tisk.cz/>
- [20] Layer thickness in 3D printing: an additive manufacturing basic. *Sculpteo* [online]. Villejuif, c2009-2017 [cit. 2017-05-24]. Dostupné z: <https://www.sculpteo.com/en/glossary/layer-thickness-definition/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CNC	číslicově řízený stroj
DDA	digitální diferenciální analyzátor
JSON	formát určený pro výměnu dat
HAL	vrstva hardwarové abstrakce
DPS	deska plošných spojů
NURBS	neuniformní racionální B-spline
G-kód	způsob zápisu trajektorie obráběcích strojů
PSD	diskrétní regulátor s proporcionální, sumační a diferenční složkou

SEZNAM PŘÍLOH

A	Obsah přiloženého CD	53
B	Vybrané řídicí příkazy	54
C	Trajektorie kruhových interpolačních algoritmů	55
D	Schéma řídicí desky	56
D.1	Schéma řídicí desky hlavní část	56
D.2	Schéma řídicí desky vstupní obvody	57
D.3	Schéma řídicí desky výstupní obvody	58
D.4	Schéma řídicí desky drivery	59

A OBSAH PŘILOŽENÉHO CD

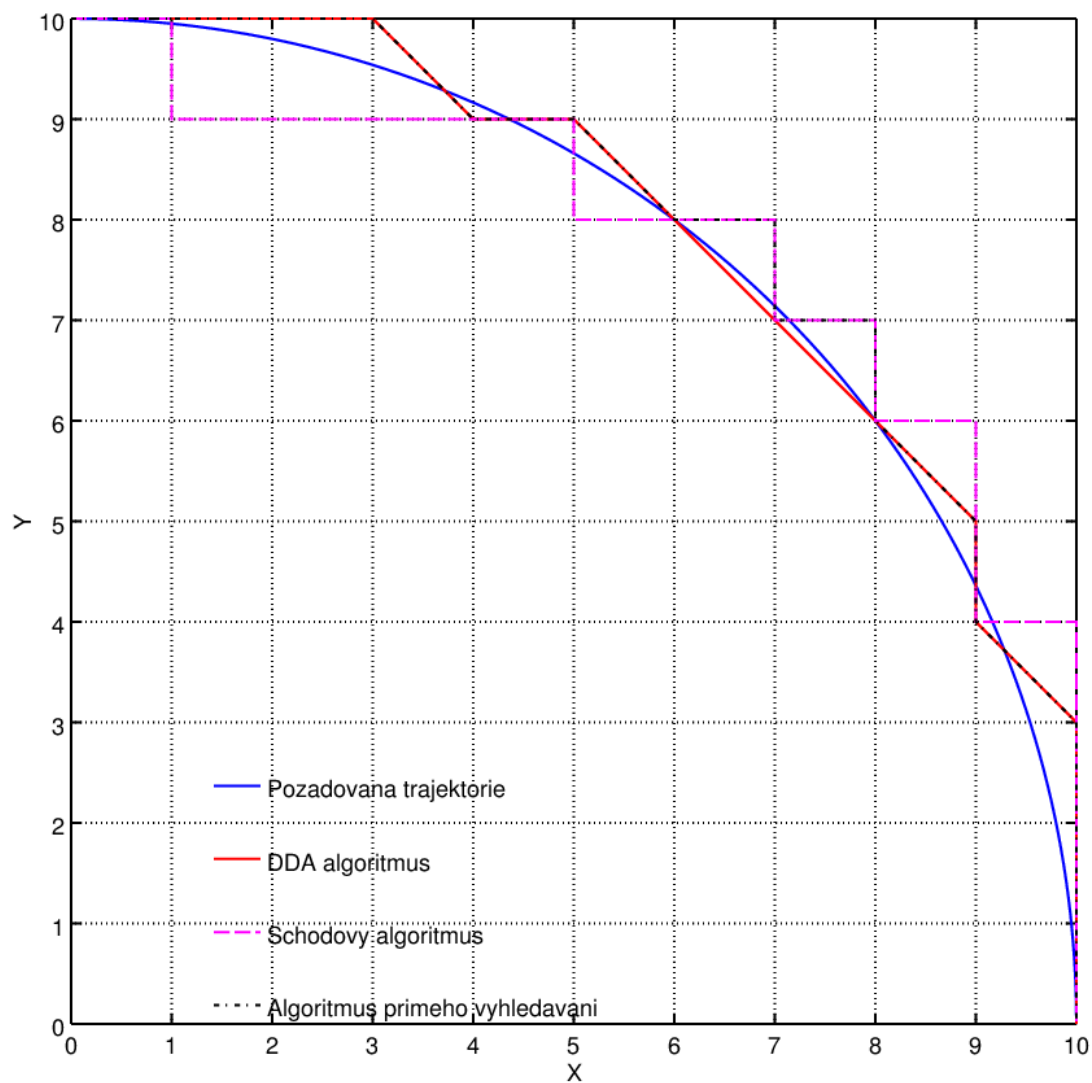
/	
└─ bakalarska_prace_Vitek.pdfElektronická verze písemné části práce
└─ zdrojove_kody_firmware.zipZdrojové kódy firmwaru pro kit Nucleo
└─ zdrojove_kody_PC_aplikace.zipZdrojové kódy PC aplikace
└─ DPS.zipSchéma a DPS jako projekt programu Kicad
└─ schema.pdf Vyexportované schéma
└─ DPS_foto.jpg Fotografie výsledné DPS

B VYBRANÉ ŘÍDICÍ PŘÍKAZY

Tab. B.1: Vybrané podporované kódy a jejich význam

Příkaz	Parametry	Význam	Příklad
G0	X/Y/Z	rychloupřesun	G0 X50
G1	X/Y/Z F	pracovní posuv	G1 X10 F100
G4	P	čekání	G4 100
G20	-	souřadnice v inch	G20
G21	-	souřadnice v mm	G21
G28	X/Y/Z	posuv do výchozí polohy	G28 X Y
G90	-	absolutní pozicování	G90
G91	-	relativní pozicování	G91
G92	X/Y/Z/E	nastavení pozice bez pohybu	G92 X5 Y10
M0	-	zastavení po ukončení probíhající věty	M0
M17	-	povolení všech motorů	M17
M18	-	zakázání všech motorů (volný pohyb)	M18
M101	T	pohyb nástroje vpřed	M101 T1
M102	T	pohyb nástroje vzad	M102 T2
M106	P/S	nastavení rychlosti ventilátorů	M106 P1 S255
M107	-	vypnutí ventilátorů	M107
M109	T/S	nastavení teploty T1/2	M109 T1 S100
M190	S	nastavení teploty T3 s čekáním	M190 S60

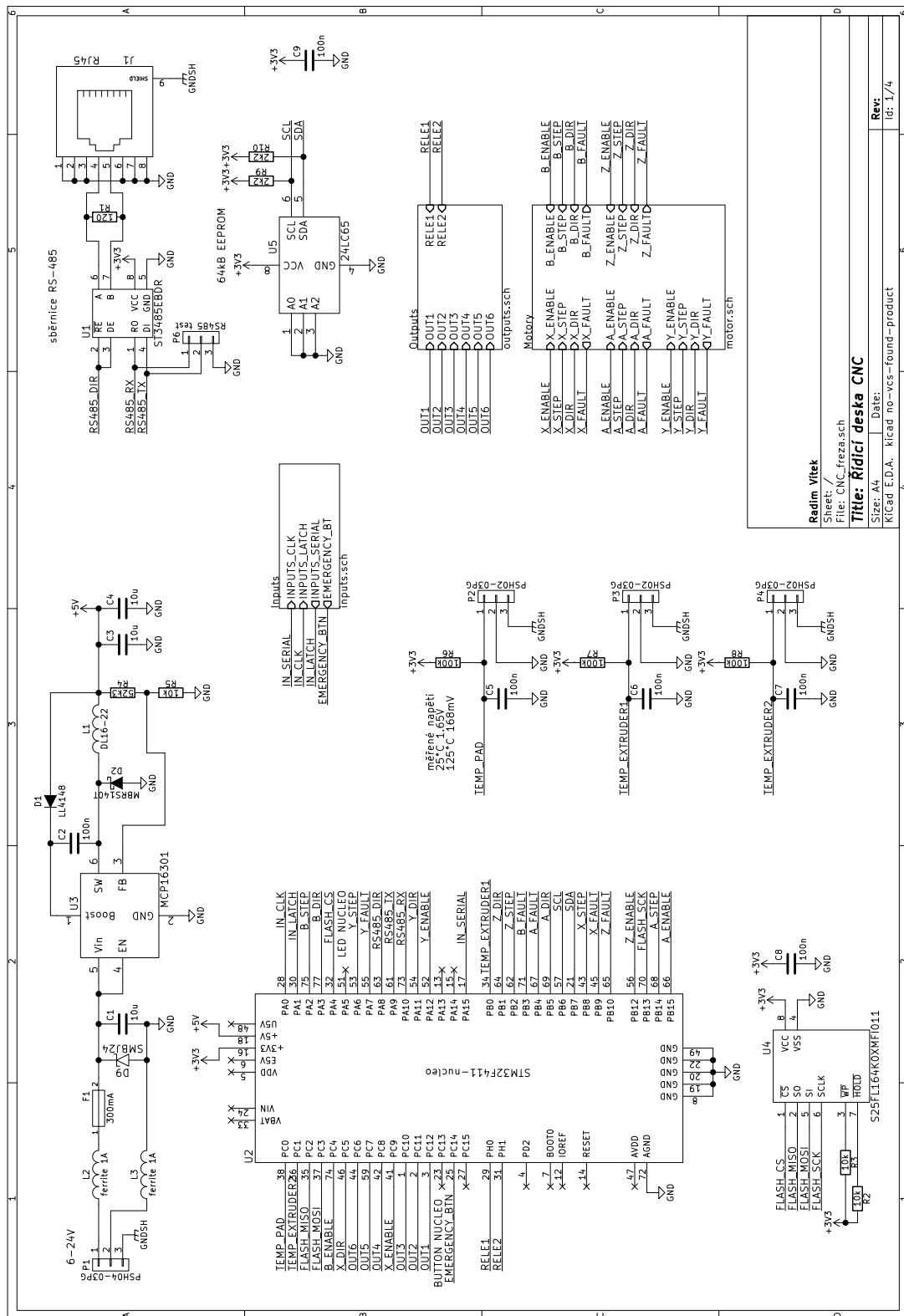
C TRAJEKTORIE KRUHOVÝCH INTERPOLAČNÍCH ALGORITMŮ



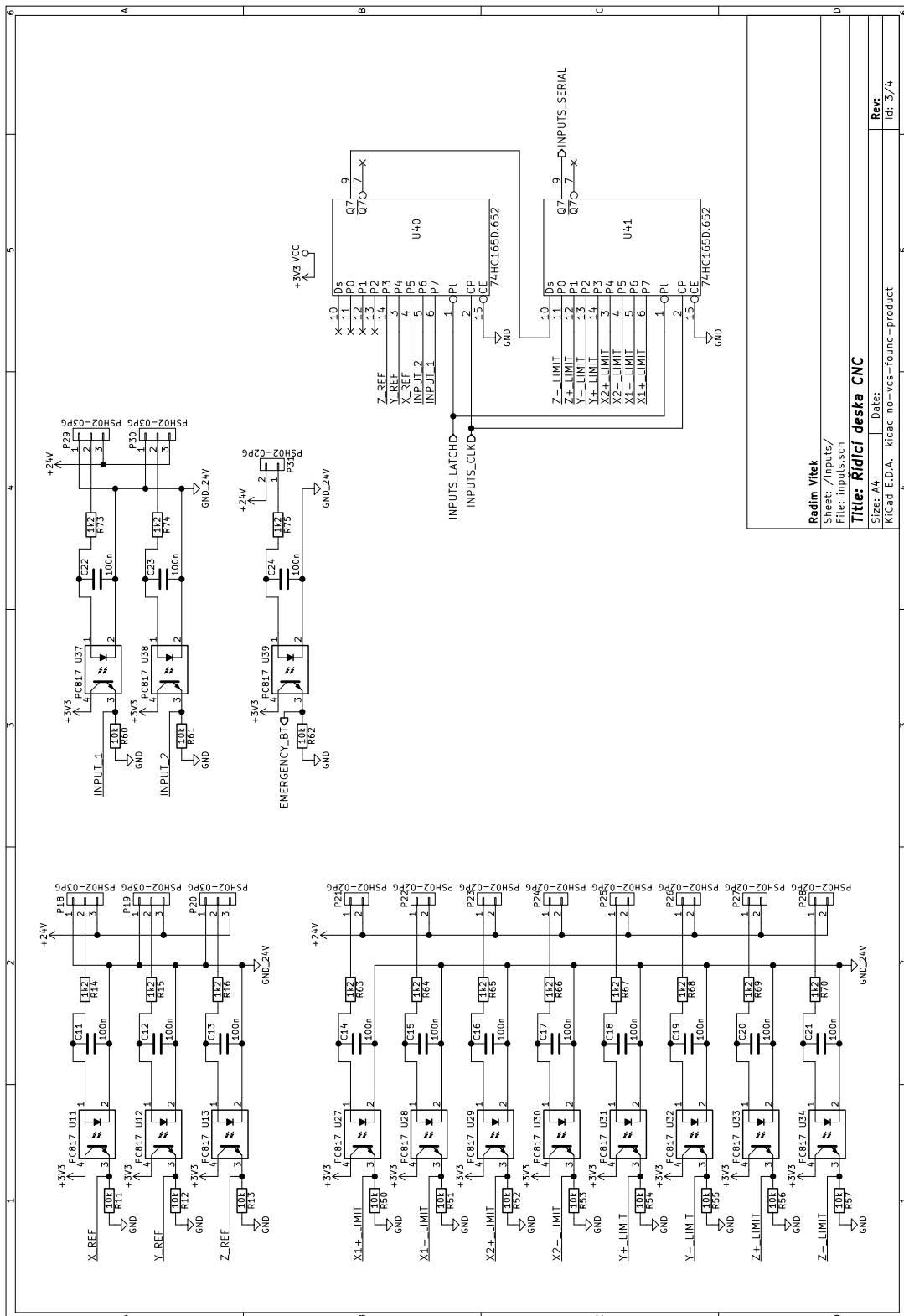
Obr. C.1: Výsledné trajektorie kruhových algoritmů

D SCHÉMA ŘÍDICÍ DESKY

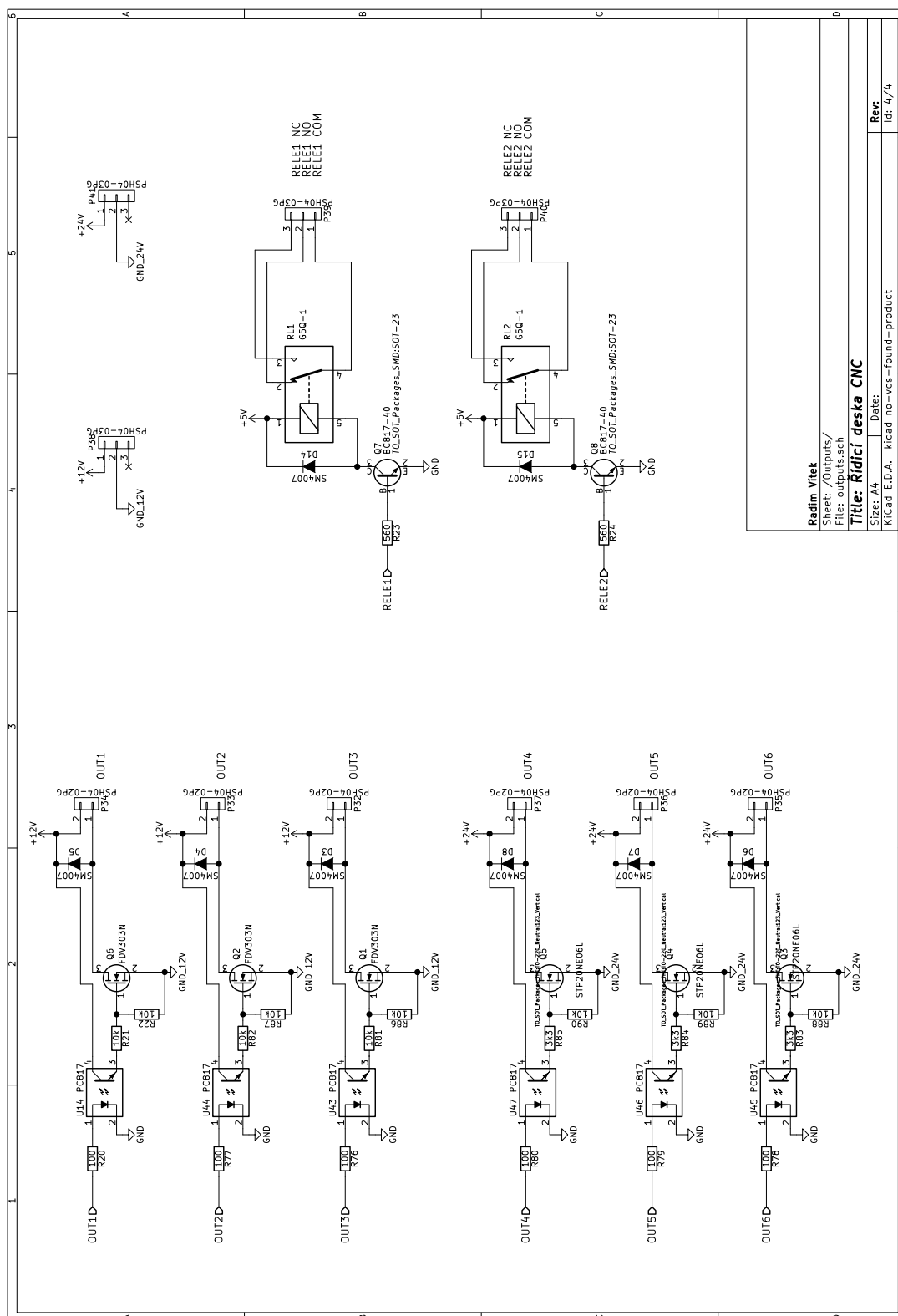
D.1 Schéma řídicí desky hlavní část



D.2 Schéma řídicí desky vstupní obvody



D.3 Schéma řídicí desky výstupní obvody



D.4 Schéma řídicí desky drivery

